UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR

TRABAJO FIN DE GRADO

DISEÑO E IMPLEMENTACIÓN DE UNA APLICACIÓN WEB PARA UNA RED SOCIAL DEPORTIVA

Vicente Teruel Rodríguez

MAYO 2015

DISEÑO E IMPLEMENTACIÓN DE UNA APLICACIÓN WEB PARA UNA RED SOCIAL DEPORTIVA

AUTOR: Vicente Teruel Rodríguez

TUTOR: Alejandro Sierra Urrecho

Dpto. Ingeniería Informática

Escuela Politécnica Superior

Universidad Autónoma de Madrid

Mayo 2015

Resumen

El objetivo de este proyecto es el diseño e implementación de Sociport, una aplicación web para crear y gestionar una red social deportiva en Internet. El propósito principal de la aplicación es facilitar el encuentro entre usuarios con perfiles deportivos en común a través de organización de eventos y creación de publicaciones.

Para ello, se ha llevado a cabo un estudio de las funciones principales que debe tener un sistema de estas características y se ha realizado un diseño modulado de la aplicación en el servidor. El conjunto de funcionalidades del sistema se encuentra en una API basada en el protocolo RPC que contiene el único punto de conexión con la base de datos relacional del sistema. Por otra parte, se encuentra la parte de aplicación que contiene las vistas del sistema y que es totalmente independiente del módulo que contiene la API. La comunicación entre el cliente y servidor se realizada mediante el protocolo HTTP y llamadas AJAX.

La aplicación utiliza un sistema de infraestructura LAMP para definir el servidor web. Para el desarrollo de la herramienta se han utilizado los frameworks de desarrollo web y de diseño Zend, Backbone y Bootstrap para facilitar la implementación de la misma. Finalmente, se han realizado un conjunto de pruebas unitarias y de integración para verificar y validar el sistema obteniendo unos resultados altamente satisfactorios. Además, distintos usuarios han evaluado muy positivamente tanto la usabilidad como las funciones del sistema. La versión resultante de este trabajo realizado íntegramente por el estudiante Vicente Teruel Rodríguez es una primera versión de la aplicación totalmente operativa.

Palabras clave

Sociport, API RPC, parte de aplicación, base de datos, LAMP, HTTP, AJAX, frameworks de desarrollo.

Abstract

The objective of this project is the design and the implementation of Sociport, a web tool designed to create and manage a social network in Internet. The principal purpose of the application is thought to facilitate the meeting between users with similar sporty profiles with organization of events and creation of publications.

First, a study of the principal functions that a system with similar features must have has been elaborated and a modulate design of the application has been realized in the server. A set of functionalities of the system are in an API based in the RPC protocol. It has the unique connection point with the relational database of the system. By the other hand, the application module contains the views system and it is totally independent of the API module. The communication between the client and the server has been realized through the HTTP protocol and AJAX calls.

The application uses a system infrastructure LAMP to define the web server. Zend, Backbone and Bootstrap, development web and design frameworks, have been utilized to ease the implementation. A set of unity and integrity tests has been performed to verify and validate the system and, finally, highly satisfactory results have been obtained. Also, different users have realized a very positive evaluation of the usability and functions of the system. The resulting version of this project integrally realized by the student Vicente Teruel Rodríguez is a first totally operative version of the application.

Key words

Sociport, API RPC, application module, database, LAMP, HTTP, AJAX, development frameworks.

Agradecimientos

Quiero agradecer a todas las personas que han hecho posible que mi aventura en la universidad haya llegado hasta aquí. A mi tutor Alejandro, por darme tantas facilidades desde la primera conversación sobre el TFG que tuvimos, animándome a seguir adelante en todo momento. Gracias también a quienes me han ayudado, motivado y dado ese empujón que tanto necesitaba en los momentos más importantes. A mi familia, que me animó a continuar en aquel primer año de carrera tan complicado y que tiene gran culpa de todo lo que estoy consiguiendo. Para terminar, quiero dar las gracias a esa pequeña familia de cinco que hemos formado en estos años y que me llevo para toda la vida. GRACIAS.

Índice de Contenidos

Índice de Figuras

Glosario

PHP: Lenguaje de programación del lado del servidor diseñado para el desarrollo web.

Javascript: Lenguaje de programación interpretado utilizado, principalmente, en el lado del cliente para el desarrollo web dinámico.

Jquery: Biblioteca de Javascript.

HTML: *HyperText Markup Language*. Lenguaje de marcado para elaboración de páginas web.

SQL: *Structured Query Language*. Lenguaje declarativo de acceso a base de datos relacionales.

HTTP: *Hypertext Transfer Protocol*. Protocolo usado para acceder a la web.

AJAX: *Asynchronous JavaScript And XML*. Técnica de desarrollo web que implementan aplicaciones interactivas.

JSON: *JavaScript Object Notation.* Estándar basado en texto plano para el intercambio de información.

Framework: Estructura de soporte definida utilizada para organizar y desarrollar un proyecto software.

API: *Application Programming Interface.* Interfaz de programación entre componentes software.

RPC: *Remote Procedure Call.* Protocolo utilizado en aplicaciones distribuidas cliente-servidor en el que no hay que preocuparse por la comunicación entre ambos.

MVC: *Model - View - Controller.* Patrón de diseño software que propone separar el código de los programas por responsabilidades.

LAMP: *Linux – Apache – MySQL – PHP.* Sistema creado por el conjunto de aplicaciones libres Linux, Apache, MySQL y PHP.

Front-end: Parte del software que interactúa con los usuarios.

Back-end: Parte del software que procesa la entrada desde el front-end.

1 Introducción

1.1 Marco del proyecto

Este Proyecto de Fin de Grado ha sido realizado por el estudiante Vicente Teruel Rodríguez partiendo de una propuesta personal del propio alumno. Dicho trabajo, tiene como propósito el diseño e implementación de una primera versión de una web deportiva mediante el uso de herramientas avanzadas de programación web.

1.2 Motivación del proyecto

En la sociedad en la que vivimos cada vez hay más gente aficionada al deporte y éste ya forma parte de nuestras vidas. En multitud de ocasiones una persona quiere practicar un deporte pero no tiene compañeros con quien hacerlo o sus horarios no coinciden con los de sus conocidos.

Actualmente, existen algunas webs que realizan esta función de unir gente interesada en los mismos gustos deportivos. Muchas de ellas estas orientadas a la competición y otro importante grupo tienen un amplio número de funcionalidades que hacen que la interacción del usuario con la web no sea sencilla. Además, se está comprobando en los últimos años que, dejando al margen grandes redes sociales como Facebook, Google +, etc., los sistemas que están teniendo mayor éxito son los que tienen un fin muy concreto, fáciles en cuanto a usabilidad y con una curva de aprendizaje muy rápida.

Por lo tanto, la motivación por la que se va a desarrollar este proyecto es construir una aplicación para enlazar aficionados con gustos deportivos en común, dando a conocer nuevas tendencias deportivas, sin discriminar a usuarios que no conocen bien las nuevas tecnologías.

1.3 Objetivo

El objetivo de este Proyecto de Fin de Grado es la definición, implementación y validación de un sistema informático web cuyo propósito es gestionar una red social deportiva en Internet para facilitar el encuentro entre todo tipo de aficionados del deporte.

El resultado de este trabajo es, por tanto, la construcción de *Sociport*, una herramienta desarrollada para web que facilita el encuentro entre individuos con gustos deportivos en común y que ha sido ideada, analizada, diseñada e implementada íntegramente por el estudiante. El sistema se visualiza a partir de una interfaz gráfica compatible con todos los navegadores modernos y se estructura en dos módulos independientes con el fin de crear un sistema cuyos servicios sean compatibles con otro tipo de soportes como, por ejemplo, dispositivos móviles. Además, se pretende acercar la aplicación al mayor número de clientes posibles ya que un aficionado al deporte no tiene por qué tener conocimientos medio-avanzados sobre navegación web y, por tanto, se pretende que la interacción del usuario con el sistema sea lo más simplificada posible. Por otro lado, se quiere dar a conocer a la sociedad una serie de deportes menos extendidos para ampliar el rango de deportes practicados entre la población.

Las funcionalidades más destacadas del proyecto son la organización de eventos deportivos entre usuarios, publicación de comentarios y noticias en muro personal, disponibilidad de una lista de actividades deportivas en las que los usuarios puedan inscribirse, creación de perfiles personales y deportivos por niveles de práctica, seguimiento entre usuarios para crear lazos en la comunidad y valoración de contenidos y deportes.

Finalmente, destacar que el sistema está conectado a una base de datos MySQL. El sistema está diseñado para ejecutarse en dos servidores totalmente independientes pero, por no ser necesario para la creación de una primera versión del programa, ambos servidores se encuentran alojados en el ordenador personal del estudiante.

1.4 Estructura del documento

La memoria está formada por las siguientes secciones:

En la S**ección 2** se hace un análisis del estado del arte. En este estudio se investigan los sistemas similares al propuesto en este trabajo, es decir, redes sociales deportivas en internet o páginas web con un punto de encuentro o materia en común entre usuarios. Este

punto es fundamental en este proyecto para comprobar, sobre todo en webs ampliamente aceptadas, las ventajas e inconvenientes que proporcionan cada una de ellas.

En la **Sección 3** se detallan los objetivos y funcionalidades buscados en este trabajo de fin de grado. En primer lugar, se detalla el objetivo principal de forma global para situar al lector en un escenario y, en segundo lugar, se describen las funcionalidades y los objetivos específicos necesarios para que el sistema se desarrolle con éxito.

En la **Sección 4** se define el sistema a desarrollar donde se describe la metodología seguida y las herramientas utilizadas para el desarrollo del mismo.

En la **Sección 5** se realiza el análisis y diseño del sistema. En la primera parte de análisis se describen los requisitos funcionales y no funcionales. En la segunda parte de diseño se describe la arquitectura y las estructuras de datos de la aplicación. Además, se explican las funcionalidades del sistema de forma detallada.

En la **Sección 6** se muestra la implementación de este trabajo. Se detalla el proceso de desarrollo de toda la aplicación junto con la base de datos y la interfaz gráfica.

En la **Sección 7** se describen las pruebas desarrolladas en la validación y verificación de la web para comprobar el correcto funcionamiento de la misma. Seguidamente, en la **Sección 9,** se evalúa el resultado final para determinar los beneficios que se han obtenido del proyecto creado.

En la **Sección 10** se describe las conclusiones obtenidas al final del proyecto y posibles líneas de trabajo futuras. Por último, en la **Sección 11** se enumeran las referencias utilizadas en el desarrollo de esta memoria.

2 Estado del arte

2.1 Introducción

Tradicionalmente, a la hora de crear nuevas relaciones sociales, las personas han establecido relaciones con los beneficiarios gracias a la presencia de intermediarios, independientemente de cuál sea el vínculo de unión. Estas relaciones requerían que tanto los individuos participantes en la relación como los intermediarios tuvieran una participación activa en el proceso.

En los últimos años ha proliferado la creación de un software tipo que ayuda en el proceso de generación de relaciones sociales entre los integrantes de una determinada población. Este producto está cada vez más en auge debido a la gran revolución que están viviendo las nuevas tecnologías y que, en su mayor parte, nos facilita el desarrollo como sociedad.

Si partimos del actual ritmo de vida que llevan las grandes ciudades y de la aglomeración de población que sufren, se entiende que necesitemos herramientas que nos den posibilidades de contactar y crear lazos con el resto de personas de la población de una forma más veloz. Internet es el punto clave en todo este proceso ya que nos proporciona una comunicación directa e instantánea con el resto del mundo.

Este Proyecto de Fin de Grado tiene como objetivo la creación de Sociport, una red social deportiva en internet, que tiene como fin unir a todo de tipo de público, independientemente de sus conocimientos sobre navegación web, a través de una única afición como es el deporte y fomentar actividades deportivas innovadoras y poco conocidas. Para crear el mejor servicio posible, se va a realizar un análisis de un conjunto de aplicaciones web distribuidas en España cuyo objetivo sea similar al previamente definido y, especialmente, que tengan un tema específico como vínculo de unión. Del estudio han quedado excluidas las redes sociales más extendidas ya que no se pretende evaluar sus funcionalidades o servicios.

2.2 Sistemas existentes

A continuación, se analizan las redes sociales que tienen como materia principal el deporte:

Timpik

La red social deportiva más extendida en España en este momento. La web intenta conectar personas para participar en eventos deportivos utilizando geolocalización, agrupar a gente para organizar campeonatos y compartir entre los usuarios del sistema resultados y estadísticas sobre sus hitos deportivos. [1]

Timpik ha sido elegida para el análisis por su amplia repercusión en el mundo del deporte y por su avanzado módulo de eventos deportivos enfocado a la competición.

Este sistema, desarrollado tanto para web como para IOS Y Android, tiene como objetivo principal la organización de eventos, partidos y torneos deportivos. En su afán organizador se desarrolló una aplicación complementaria llamada Timpik PRO que está creada para superorganizadores que necesitan una plataforma con algunos extras muy exigentes. Por ello, tiene dos formas de registro en la web: versión usuario y versión organizador.

Timpik tiene un módulo que posibilita la reserva de pistas directamente desde la propia web a través de un monedero electrónico que se puede recargar desde una tarjeta de crédito/débito o cuenta bancaria. También cuenta con otras opciones como creación o participación en clubs virtuales, lista de espera en eventos que ya se encuentran con todas las plazas ocupadas, foros de partidos, estadísticas sobre logros y asignación de medallas, notificaciones y amistades entre usuarios, disponibilidad de una oferta de más de 80 deportes, etc.

Todas estas funcionalidades se vuelven, en cierto punto, en contra de la propia web ya que hace que la interacción del usuario con la misma no sea muy intuitiva para usuarios con bajos e incluso medios conocimientos sobre navegación web. Este hecho acota de forma considerable el público que puede acceder a la red de usuarios y que busca una herramienta más sencilla para conocer gente del mundo del deporte.

Decathlon Sport Meeting

Nos encontramos con una de las redes sociales más conocidas en el ámbito nacional debido a la marca que la da nombre: Decathlon. Decathlon Sport Meeting (DSM) es una comunidad que permite encontrar personas que compartan gustos deportivos y la pasión por el deporte [2].

Este sistema web ha sido propuesto para el estudio por su fuerte semejanza en los objetivos principales con nuestra aplicación. Su importante enfoque de red social, añadido al alto nivel de accesibilidad que posee la web, hace que DSM se convierta en una herramienta válida para todo tipo de usuarios.

Creación de un perfil deportivo, organización sencilla de eventos por niveles de dificultad atendiendo a la técnica deportiva, opción de seguimiento de usuarios, creación de listas de deportistas favoritos, invitaciones vía e-mail, Twitter o Facebook para publicar propuestas al exterior de la red o sistema de publicaciones en muro personal son algunas de las funcionalidades que destacan de la web [2]. A diferencia de Timpik, esta web no está tan orientada al mundo de la competición sino que se acerca más a la idea de conocer y establecer lazos deportivos entre los usuarios. También está disponible para dispositivos móviles.

Por otra parte, la web parece encontrarse permanentemente en una versión beta. Desde hace aproximadamente un año se ofertan una serie de deportes que a la hora de ser elegidos aparecen como no disponibles para el usuario. Además, en la especificación de los deportes que un usuario puede añadir a su perfil, no se proporciona descripción ninguna del mismo ni imágenes que puedan ser de atractivo para el cliente.

Xporty

Xporty es una red social deportiva orientada, principalmente, a la organización de eventos y torneos deportivos en los diferentes municipios españoles y a la unión de individuos con deportes en común [3]. Actualmente cuenta con un total de

20.100 usuarios en su plataforma.

Xporty ha sido elegida para este análisis ya que se encuentra entre las cinco páginas más usadas en España en cuanto a redes deportivas.

Este sistema incorpora secciones comunes con las anteriores webs como organización de eventos, creación de perfil deportivo, publicaciones, etc. y, además, otras como instalaciones, equipos, grupos y torneos ya que está más orientado a la competición deportiva. Este es su fin más destacado por encima de conocer usuarios desde la propia web. Dispone de un módulo separado solo de torneos que actualmente dispone de una versión 2.0 en fase beta que permite organizar ligas, rankings, eliminatorias, etc. La app posee dos formas de registrarse en la aplicación: como usuario y como organizador.

Para ser una web que intenta unir a deportistas y no a usuarios experimentados en web, hay una serie de requisitos de usabilidad básicos que no se cumplen. Aunque la organización global del sitio está diseñada pensando en los contenidos, la navegación no ofrece siempre caminos claros y la curva de aprendizaje de la interfaz no se consigue de manera rápida, desplazando a usuarios sin conocimientos medios y avanzados de web.

Deportmeet

Deportmeet, nacida en 2011, es una red social generalista para aficionados al deporte que permite buscar gente con aficiones en común [4]. Actualmente, se encuentra en expansión y nombres importantes del mundo del deporte recomiendan el uso de esta herramienta como vínculo de unión y como plataforma para dar a conocer deportes innovadores o con menos repercusión social.

Esta red social ha sido elegida para el estudio debido a su constante auge en los últimos cuatro años, al amplio catálogo de deportes que ofrece y por la incorporación de una serie de servicios innovadores.

Deportmeet oferta servicios comunes con otras redes como organización de eventos, seguimiento de perfiles, adición de deportes, etc. y, por otro lado, ofrece una serie de servicios novedosos como la creación de una plataforma de integración y de promoción de deportes minoritarios y orientados a individuos con diversidad funcional, acceso a contenidos y artículos realizados por referentes del deporte español y periodistas deportivos o sistema de compra-venta de artículos entre usuarios de la plataforma.

En esta segunda parte del análisis se van a detallar redes sociales que tienen un vínculo en común entre usuarios sin ser, necesariamente, un vínculo deportivo:

SocialDrive

Esta aplicación, únicamente disponible para móvil, es una red social de conductores en tiempo real a través de la cual se puede informar y ser informado sobre posibles incidencias en la carretera en todos los puntos a nivel estatal [5].

SocialDrive ha sido analizada debido a la fuerte línea de crecimiento de usuarios que ha sufrido en estos meses ya que, actualmente, cuenta con más de un millón de descargas. Ver el por qué de tan rápida aceptación entre los usuarios teniendo en cuenta que en el mercado se ofertan otras aplicaciones similares tiene una fácil respuesta: su sencillez de uso y unos objetivos muy claros y definidos, como es avisar y ser informado de incidencias en el tráfico, hacen que un amplio rango de usuarios opten por utilizar esta aplicación para móvil.

Filmaffinity

Esta aplicación es una página web de películas y series que además incorpora un sistema de votación y recomendación de películas muy potente que, en la actualidad, cuenta con más de 500.000 usuarios en activo [6].

Por dicho sistema de votación que incorpora se ha añadido a Filmaffinity en el conjunto de webs a analizar. Tratar de guiar a un usuario a partir de las opiniones de otros anónimos y no de los administradores de la propia web, hace que un cliente se sienta más seguro a la hora de elegir contenidos independientemente del tema propuesto.

Facebook

Facebook, como red social por excelencia, va siempre por delante en cuanto a servicios y funcionalidades. Para acotar el estudio, nos centramos únicamente en su sistema de likes: este servicio de votación, divertido para los usuarios, indica que contenidos destacan por encima del resto [7].

2.3 Conclusiones

En el punto anterior se han analizado los servicios existentes que comparten objetivos similares a los planteados para el proyecto. Como se ha visto, se trata de sistemas web deportivos en los que el usuario no entiende de manera clara el contenido y la navegación no se hace de forma cómoda y sencilla. Posiblemente, al intentar ofrecer un gran número de servicios avanzados hace que un sector de clientes, con conocimientos bajos o medios de navegación web, queden desplazados en el uso de dichas páginas. Estas carencias hacen que sea necesario un sistema web más usable y con unos servicios mucho más definidos para unir a todo tipo de usuarios aficionados al deporte.

Además, a partir del estudio se han extraído las siguientes conclusiones:

i. La importancia de definir claramente los servicios principales que va a ofertar nuestro trabajo sin necesidad de añadir alrededor de los mismos un gran número de servicios secundarios que distraigan la atención de los usuarios.

ii. Se busca unir a un conjunto de población que comparta una serie de gustos sin conocer, previamente, sus conocimientos sobre navegación web. La forma de intentar conseguir un amplio rango de clientes es creando una aplicación accesible para la mayoría posible. La adición de tutoriales sobre cómo usar la web puede ser una buena forma complementaria para facilitar el uso de la misma.

iii. Para crear una comunidad de usuarios e intentar atraer a sectores de la población a determinados deportes innovadores o poco conocidos se deben incluir secciones informativas sobre dichos deportes y no dar por supuesto que el cliente ya los conoce.

iv. Los sistemas de recomendación en las aplicaciones es la mejor forma de guiar a los usuarios a los contenidos que más le podrían interesar. Incorporar sistemas de votación de contenidos puede ser una solución óptima para aconsejar a los usuarios.

3 Objetivos/Funcionalidades

En este capítulo se va a explicar el objetivo principal del sistema y se van a detallar los objetivos específicos para los que se ha desarrollado la herramienta atendiendo a sus funcionalidades.

3.1 Objetivo Genérico

El objetivo de este trabajo es la definición, implementación y validación de un sistema para una red social deportiva en Internet. Esta tecnología se implementará en forma de aplicación web.

3.2 Objetivos específicos y Funcionalidades

En los siguientes puntos se van a describir los objetivos específicos y las funcionalidades de la aplicación desarrollada:

3.2.1 Organización de eventos deportivos

Este sistema permite a los usuarios crear eventos sobre un determinado deporte con la finalidad de que el resto de usuarios se unan al mismo hasta completar todas las plazas disponibles.

El usuario creador del evento pasará a ser el organizador y, además, será incluido con el rol de participante. Dicho organizador deberá establecer unos parámetros informativos obligatorios como fecha del evento, lugar de encuentro, descripción, número máximo de participantes, etc. Dependiendo del número de participantes y la fecha actual, el evento puede encontrarse en 3 estados diferentes: abierto, completo o cerrado.

Los participantes adjuntos al evento pueden salirse en cualquier momento dejando su plaza libre para otros posibles usuarios. El organizador no puede abandonar el evento, únicamente tiene la potestad de cancelarlo, eliminándose de la web.

3.2.2 Publicaciones en tablón personal

La aplicación permite la escritura de publicaciones, formadas por un título y una descripción, en el muro personal de cada usuario para ser vistas por el resto de la comunidad teniendo que ser indicado el deporte con el cual se relaciona la publicación.

A diferencia de los eventos, las publicaciones tienen únicamente carácter informativo para el resto de la red social.

3.2.3 Inscripción de deportes

El sistema oferta más de 25 deportes para que los usuarios de la red puedan añadirlos a su perfil. Una vez inscrito un determinado deporte, el usuario tiene acceso a las dos funcionalidades anteriores, es decir, a todos los eventos y publicaciones relacionados con dicho deporte. En cualquier momento el cliente puede cancelar la inscripción a un deporte.

3.2.4 Perfiles personales y deportivos

Al tratarse de una comunidad que tiene como fin conocer nuevos usuarios con deportes en común, el sistema necesita que los usuarios creen sus propios perfiles para dar a conocerse.

En primer lugar, es necesario establecer un perfil con los datos personales del usuario. Dichos datos se dividen en dos secciones:

- Datos obligatorios: necesarios para poder registrarse en el sistema y poder tener acceso al mismo.
- Datos opcionales: dirigidos a que otros usuarios puedan conocer mejor a un individuo.

Por otra parte, por cada deporte que un beneficiario inscribe, se tiene que crear un perfil deportivo individual en el que se informe de una serie de datos como nivel de juego, lugar de desarrollo, días y horas que se lleva a cabo su práctica, etc. De esta forma, se proporciona información muy detallada de cada actividad deportiva, simplificando la búsqueda de usuarios con gustos en común.

3.2.5 Seguimiento de usuarios

La web dispone de un sistema de amistades en el que un usuario puede ‘seguir’ a cualquier individuo de la comunidad y, a su vez, puede ser ‘seguido’ por otros.

De esta forma, si se tienen aficiones en común con un usuario y se quiere conocer que publica o que eventos organiza, en el timeline aparecerán las notificaciones del amigo en cuestión. La web dispone de una sección de amistades con un mayor número de opciones relacionadas con esta funcionalidad.

3.2.6 Valoración de contenidos

Tras el estudio del estado del arte se comprobó que un usuario se guía por las valoraciones que realizan otros usuarios de la comunidad sobre diversos temas. Por consiguiente, el sistema permite realizar una valoración numérica de otros usuarios de la comunidad y de los deportes en los que se encuentre inscrito. Para ello, se utiliza una ventana con cinco estrellas.

Por otra parte, los eventos y publicaciones disponen de un sistema de ‘MG’ para ver que contenidos son más destacables. Un usuario solo puede insertar un ‘MG’ por cada contendido.

3.2.7 Control de usuarios y contenidos

El acceso al sistema se puede realizar como dos tipos de usuario: cliente y administrador. El administrador dispone de una serie de servicios para mantener la red social limpia de contenidos no apropiados, pudiendo incluso eliminar usuarios de la red.

4 Definición del Proyecto

4.1 Metodología

El desarrollo de este TFG ha seguido un ciclo de vida en cascada con realimentación. Se ha elegido este ciclo de vida ya que, dado el tamaño de la aplicación, ofrece la posibilidad de realizar diferentes cambios en el conjunto de fases durante todo el ciclo de vida software [8]. De esta forma, ha permitido realizar ciertas modificaciones en fases ya concluidas con el fin de mejorar funcionalidades durante el desarrollo del proyecto. Por otra parte, no se tenía la certeza que la estimación de tiempos realizada en el inicio pudiese cumplir los plazos, por lo que iba a ser necesario retroceder en alguna fase para añadir o eliminar funcionalidades.

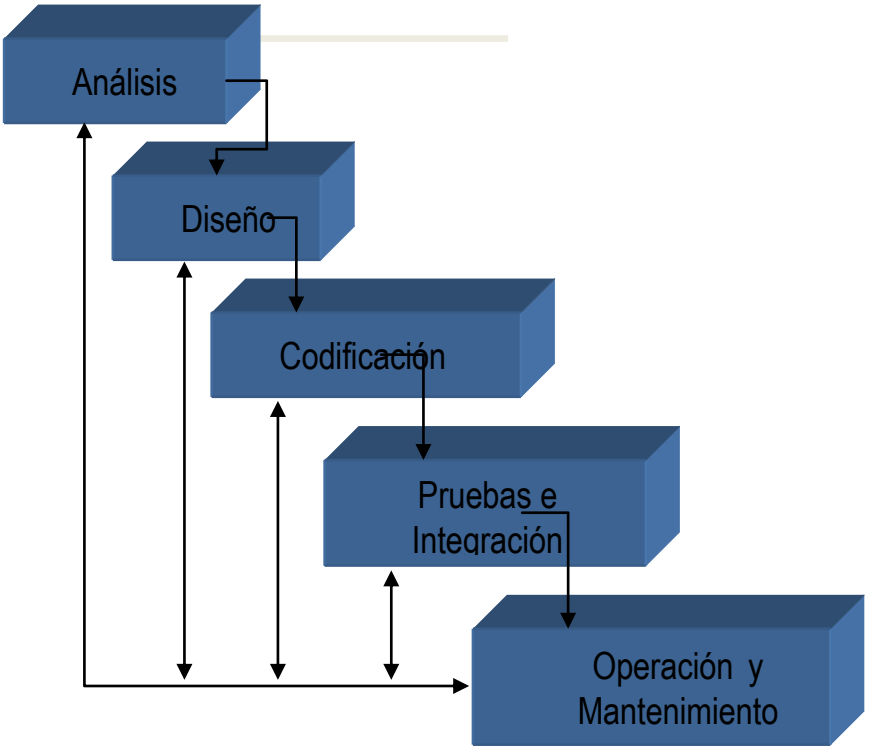

Figura 4.1: Esquema de un ciclo de vida con realimentación

A continuación, se detallan las tareas realizadas en cada una de las fases del desarrollo:

Análisis

- Análisis del estado del arte.
- Definición de objetivos y funcionalidades.
- Especificación de requisitos funcionales y no funcionales.

Diseño

- Diseño del diagrama Entidad-Relación (E-R) para la base de datos MySQL.
- Diseño de la interfaz de usuario.
- Diseño del diagrama de clases de la parte de la aplicación.
- Diseño del diagrama de clases de la API RPC.

Codificación

- Implementación de la base de datos: Mysql
- Implementación del front-end del sistema: HTML + Javascript
- Implementación del back-end del sistema: PHP

Pruebas y validación

- Realización de la estrategia de pruebas.
- Diseño de los casos de prueba.
- Desarrollo de pruebas.
- Evaluación del sistema tras el plan de pruebas.

4.2 Herramientas usadas

Para la implementación de esta aplicación se han utilizado las siguientes herramientas:

- Lenguaje para el desarrollo y de manejo de la base de datos: **MySQL**
- Lenguaje de programación web del lado del servidor: **PHP**
- Lenguaje de programación de desarrollo de interfaces web: **HTML**
- Lenguaje de hojas de estilos: **CSS**
- Lenguaje de programación interpretado: **JavaScript**
- Biblioteca de JavaScript: **Jquery**
- Framework de desarrollo de aplicaciones web: **Zend Framework**
- Framework de desarrollo de aplicaciones JavaScript: **Backbone**
- Framework de diseño web: **Bootstrap**
- Servidor web HTTP: **Apache**

Desarrollo y manejo de bases de datos: MySQL 5.1 y Workbench 6.2

Para la definición de la base de datos se ha utilizado el lenguaje SQL a través del sistema de gestión MySQL [9]. Actualmente, este sistema de gestión de bases de datos relacional es el más extendido por encima de otros como PostgreSQL, SQLite, etc. Se ha usado la versión 5.1 para Linux.

Para facilitar la interacción con la base de datos se ha utilizado una herramienta visual llamada Mysql Workbench. Dicho software permite crear, diseñar, administrar y mantener una base de datos MySQL [10]. La última versión disponible es la 6.2.

Lenguaje de programación: PHP 5.6

El código de este TFG en la parte del servidor o back-end se ha implementado en lenguaje PHP ya que se trata de un lenguaje especialmente adecuado para el desarrollo web [11]. PHP puede ser inscrustado en código HTML pero, salvo en mínimos puntos del proyecto, no se ha utilizado este método sino que se utiliza código JS embebido con Backbone. La versión utilizada ha sido la 5.6.

Lenguaje de programación: HTML 5

El código de las vistas de la aplicación se ha implementado en lenguaje HTML. Este lenguaje de etiquetas es un estándar a cargo de la W3C y es la base para la creación de páginas web tradicionales [12]. Se ha utilizado este lenguaje porque permite agregar scripts al código, punto clave en el desarrollo de este trabajo. HTML 5 ha sido la versión elegida para el desarrollo.

Lenguaje de hojas de estilos: CSS 3

El lenguaje CSS o Hoja de estilo en cascada ha sido elegido para definir la presentación de los documentos HTML de este trabajo. De este forma, separamos el estilo y el contenido en las vistas, estructurando el código de manera más óptima [13]. La versión utilizada ha sido la 3.0.

Lenguaje de programación y bibliotecas: Javascript 1.8 y Jquery 2.1.1

La web planteada en este trabajo tiene un importante diseño dinámico. Por ello, se ha elegido Javascript (JS), un lenguaje orientado a objetos, para implementar la parte del cliente junto a HTML [14]. Por otra parte, para interactuar de forma más sencilla con el código HTML de la app se ha utilizado Jquery, la biblioteca más extendida de Javascript [15]. Todos los navegadores modernos soportan JS.

Finalmente, destacar que además de crear multitud de efectos para mejorar la interfaz de usuario (animaciones, eventos, etc.), JS se ha utilizado fundamentalmente para el intercambio de información con el servidor que contiene la API RPC mediante llamadas AJAX.

Framework de desarrollo PHP: Zend

Debido a la complejidad de programación del TFG se ha dedicido desarrollar el mismo con la ayuda de un framework. Un framework proporciona una serie de funciones, clases o librerías, organizadas bajo una estructura, para facilitar el desarrollo de un proyecto.

¿Y por qué la elección de Zend? Aunque el alumno no tenía ningún conocimiento sobre Zend y la curva de aprendizaje es más lenta que otros frameworks como Codeigniter, CakePHP o Joomla, se ha optado por el uso de esta herramienta por la buena documentación que posee en la red y por el amplio rango de empresas de programación web que utilizan este framework en su desarrollo [16].

Framework de desarrollo Javascript: Backbone 1.1.2

Desarrollar un proyecto web dinámico con abundante código Javascript en el cliente puede llegar a ser tedioso para el programador. Por este motivo, se ha optado por el uso de Backbone, un framework para construir aplicaciones usando JS siguiendo el patrón MVC o, de forma más correcta, MVR –Modelo-Vista-Router– [17]. Los elementos que forman Backbone son los siguientes [18]:

- *Models* y *Collections*: Modelos que gestionan los datos.
- *Views*: Vistas que manejan la interacción con los usuarios.
- *Routers*: Enrutadores que realizan los procesos de navegación.

Framework de diseño web: Bootstrap 3.0

La parte de diseño de la web se ha realizado con la ayuda de Bootstrap 3.0, un framework que simplifica el proceso de creación de diseños web gracias a una serie de plantillas CSS y de ficheros Javascript. Entre las principales ventajas de Bootstrap 3 destacan: compatibilidad con todos los navegadores actuales, diseño escalable para ser visualizado en diferentes dispositivos, soporte para diseños sensibles, etc. [19].

Además, se han adjuntado al proyecto los siguientes complementos adicionales para Bootstrap: Datetime [20], Datetimepicker [21], Flat UI [22], Font-awesome [23], Binary Admin [24] y Bootstrap-star-rating [25].

Servidor web: Apache 2

Como servidor por excelencia, estable y robusto, se ha elegido Apache 2 como el encargado de aceptar y gestionar la entrega de las peticiones de páginas y otros recursos que provienen de los clientes de la web [26]. El servidor se encuentra corriendo sobre el sistema operativo Linux Ubuntu 12.04.

Una vez descritas las herramientas utilizadas, se puede exponer que el sistema desarrollado para construir la aplicación utiliza un sistema de infraestructura LAMP, es decir, trabaja con el sistema operativo Linux y con las herramientas Apache, MySQL y PHP. Este paradigma de programación para el desarrollo se utiliza principalmente para servidores web.

5 Análisis y Diseño

5.1 Análisis

Una vez realizado el análisis del estado del arte y de las necesidades que les surgen a los clientes, se van a enumerar los requisitos necesarios para dotar a la aplicación de todas las funcionalidades para que el sistema tenga un comportamiento correcto.

5.1.1 Requisitos Funcionales

A continuación, se describen los requisitos funcionales:

RF. 1 *La aplicación permitirá a los usuarios darse de alta en la red mediante un conjunto de datos personales obligatorios.*

RF. 2 *El sistema permitirá loguearse a los usuarios y administradores desde puntos de entrada diferentes.*

RF. 3 *Cada cliente dispondrá de un perfil personal que podrá ser actualizado con una serie de datos personales opcionales.*

RF. 4 *El sistema ofertará una lista de deportes disponibles para los usuarios de la comunidad. La lista sólo podrá ser modificada por el usuario máster de la aplicación y no por otro rango de usuarios.*

RF. 5 *Cada cliente podrá añadir a su perfil cualquier deporte disponible ofertado por el sistema, pudiendo eliminarlo del mismo en todo momento. Será necesario indicar el nivel practicado para la inscripción de cada deporte.*

RF. 6 *Cada cliente dispondrá de un perfil deportivo por cada deporte en el que se encuentre inscrito. Este perfil podrá ser actualizado con un conjunto de datos opcionales.*

RF. 7 *El sistema permitirá a los usuarios ver el perfil personal y perfil deportivo del resto de usuarios de la red.*

RF. 8 *Cada cliente podrá 'seguir' a cualquier otro usuario y, a su vez, podrá ser 'seguido' por otros usuarios. Cuando un usuario X sigue a otro usuario Y, este último se añade a la sección amigos-siguiendo del usuario X y, al mismo tiempo, se añade a la sección de amigos-seguidores del usuario Y.*

RF. 9 *Cada usuario puede 'dejar de seguir' a otro dejando de ver sus notificaciones en el timeline y quedando eliminado de la lista de amistades.*

RF. 10 *El sistema recomendará usuarios que no se encuentren enlazados para facilitar el encuentro entre individuos con características similares tanto personales y como deportivas.*

RF. 11 *El sistema dispondrá de una sección de amistades con un conjunto de filtros para ver a los usuario de la red: siguiendo, seguidores y recomendaciones/público.*

RF. 12 *Cada usuario podrá crear un evento deportivo definiendo una serie de datos obligatorios (título, descripción, fecha y hora, lugar, número total de participantes y deporte al que pertenece) para que otros usuarios se unan al mismo. Estos datos podrán ser actualizados en cualquier momento por el organizador, es decir, el usuario creador del evento.*

RF. 13 *Cualquier usuario no organizador de un evento podrá unirse al mismo para participar siempre que no esté el cupo de personas completo o la fecha del evento sea menor que la fecha actual. Dicho usuario podrá salirse del evento en cualquier momento.*

RF. 14 *Cada usuario podrá crear publicaciones en su tablón personal indicando siempre al deporte a la que pertenece. Esta publicación podrá ser actualizada por el mismo usuario.*

RF. 15 *Todo usuario podrá borrar contenido de su tablón, tanto eventos como publicaciones.*

RF. 16 *El sistema permitirá hacer un filtrado de eventos y publicaciones en el timeline del usuario y en el tablón personal.*

RF. 17 *El sistema dispondrá de un método de evaluación por estrellas para que los usuarios puedan valorar deportes en los que se encuentren inscritos.*

RF. 18 *El sistema dispondrá de un método de evaluación MG para que los usuarios puedan valorar eventos y publicaciones de otros usuarios.*

RF. 19 *Cada usuario podrá darse de baja de la aplicación, perdiendo todo el contenido almacenado desde que se registro en la web.*

RF. 20 *El usuario máster podrá dar de alta y baja a los administradores del sistema.*

RF. 21 *Los administradores de la red podrán eliminar los siguientes contenidos de la red: cuentas de usuarios, eventos y publicaciones.*

5.1.2 Requisitos no funcionales

A continuación, se describen los requisitos no funcionales del sistema:

RNF. 1 *La aplicación dispondrá de una interfaz gráfica sencilla e intuitiva para que la interacción de los usuarios sea lo más fácil posible.*

RNF. 2 *Los datos mostrados a los administradores tendrán un diseño más complejo en forma de listas y tablas para estructurar la información de forma óptima.*

RNF. 3 *La interfaz mostrará mensajes de error si los contenidos o las respuestas del servidor son erróneas.*

RNF. 4 *El idioma disponible en el sistema será únicamente el castellano.*

RNF. 5 *Se podrá utilizar la aplicación en cualquier navegador moderno, exceptuando Internet Explorer, como Chrome, Mozilla u Opera.*

RNF. 6 *Se requiere identificación de usuario con contraseña para acceder al sistema.*

RNF. 7 *La base de datos tiene un espacio limitado (200 GB) y podrá procesar un máximo de peticiones simultáneas. El usuario máster del sistema puede ampliar el tamaño de ambos si fuese necesario.*

5.2 Diseño

Tras establecer los requisitos del sistema en el apartado anterior, en el siguiente punto se va a definir la arquitectura general del sistema y, a continuación, se va a exponer en detalle el diseño de la base de datos, la interfaz gráfica y la API RPC.

5.2.1 Arquitectura general

En la figura 5.1 se muestra la estructura de la aplicación. A continuación, se exponen los componentes principales que la componen:

- Una base de datos que cumple con el modelo relacional en la que se almacena toda la información necesaria para posteriores usos como, por ejemplo, cuentas de usuarios, perfiles deportivos de los clientes, datos relacionados con los deportes ofertados, eventos y publicaciones, etc.

- Una API basada en el protocolo RPC (Remote Procedure Call) que posee un conjunto de operaciones que dota de funcionalidad a la aplicación. Este módulo dispone del único punto de acceso a la base de datos del sistema. Está formada por controladores, modelos y una capa de negocio intermedia entre ambos. La aplicación accede a la API por mediante llamadas AJAX y, dicha API, devuelve las respuestas en formato JSON.

- Un módulo que contiene la parte de la aplicación. Esta parte es totalmente independiente de la API y no tiene conexión directa con la base de datos. Está formada por controladores y vistas. La funcionalidad de dichos controladores es mínima ya que únicamente proporcionan el código de las vistas y gestionan el paso de variables entre páginas. El acceso a este módulo se realiza mediante el protocolo HTTP. El módulo está formado por la aplicación cliente y la aplicación de administrador.

- Por último, la interfaz de usuario permite tanto al cliente como a los administradores interactuar con el sistema parar realizar búsquedas de usuarios con gustos comunes, añadir deportes a un perfil deportivo, organizar eventos, escribir publicaciones sobre actividades deportivas, controlar el contenido de la web, etc.

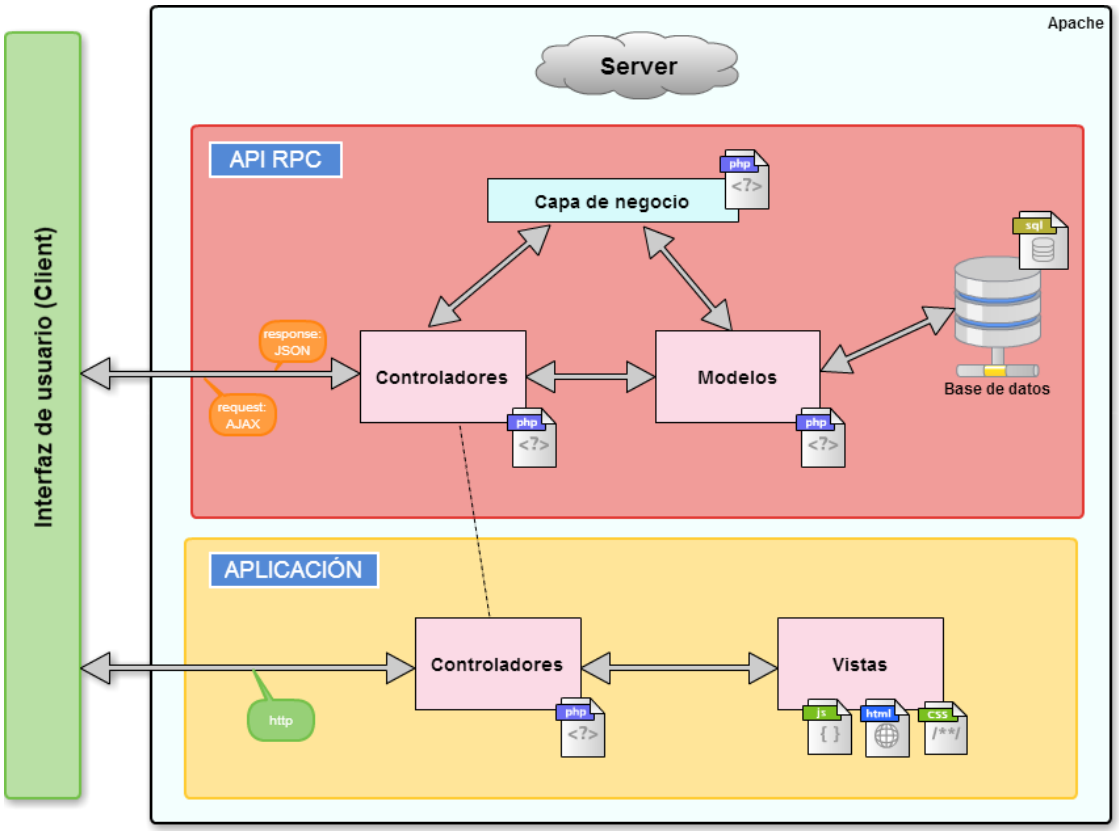

Figura 5.1: Diagrama de la arquitectura general de la aplicación

5.2.2 Interfaz de usuario

El diseño de la interfaz de usuario se ha realizado partiendo del objetivo de que la mayor parte de clientes tengan una interacción lo más intuitiva posible con la aplicación, es decir, se ha realizado un diseño centrado en el usuario. Los principios fundamentales por los que se ha guiado el diseño son los siguientes: familiaridad del usuario, consistencia, mínima sorpresa y recuperabilidad.

5.2.3 Base de datos

Para que persistan los datos en la aplicación se ha utilizado una base de datos relacional que almacena todos los datos relacionados con deportes, niveles de práctica, usuarios, eventos, publicaciones y valoraciones. Aunque se ha seguido un ciclo de vida en cascada para posibles modificaciones en fases anteriores, se ha realizado un diseño robusto de la base de datos debido a la importancia que adquiere el sistema de almacenamiento en el proyecto.

En la figura 5.2 se muestra el diseño del diagrama E/R correspondiente a la base de datos. A continuación, se describe brevemente cada una de las entidades y relaciones que componen la base de datos:

Entidades

- **USUARIOS:** representa a los clientes que se han registrado en la aplicación. La contraseña de usuario *pass* está encriptada en md5.

- **DEPORTES:** representa los deportes que ofrece el sistema. Contiene la información básica del mismo.

- **NIVELES:** representa un determinado nivel de práctica que está asociado a un deporte único. Un deporte debe contener al menos un nivel para estar disponible en el sistema.

- **EVENTOS:** es la entidad encargada de guardar los eventos organizados por un usuario sobre un determinado deporte y ofrecidos para un conjunto de niveles [1-max]. Otros usuarios no organizadores pueden unirse al evento a través de otra relación.

- **PUBLICACIONES:** es la entidad encargada de guardar las publicaciones escritas por los usuarios sobre un determinado deporte.

- **PROVINCIAS:** es una tabla que contiene las 52 provincias del estado español distinguidas por un identificador único. Está disponible en las fuentes oficiales del estado aunque dicha tabla es un volcado a sql ya que se ofrecen en formato csv.

- **MUNICIPIOS:** es la entidad que contiene todos los 1.552 municipios del estado español distinguidos por un identificador único y por el DC. También se encuentra disponible en la webs oficiales del estado. Es un volcado de dichas fuentes ya que se ofrecen en formato csv.

- **TUTORIALES:** es una tabla que almacena el contenido de una guía de ayuda que se ofrece al usuario para aprender a interactuar con el sistema.

- **ADMINISTRADORES:** es la entidad que contiene los administradores que tienen acceso a las operaciones de control y mantenimiento de la red. La contraseña de administrador *pass* está encriptada en md5.

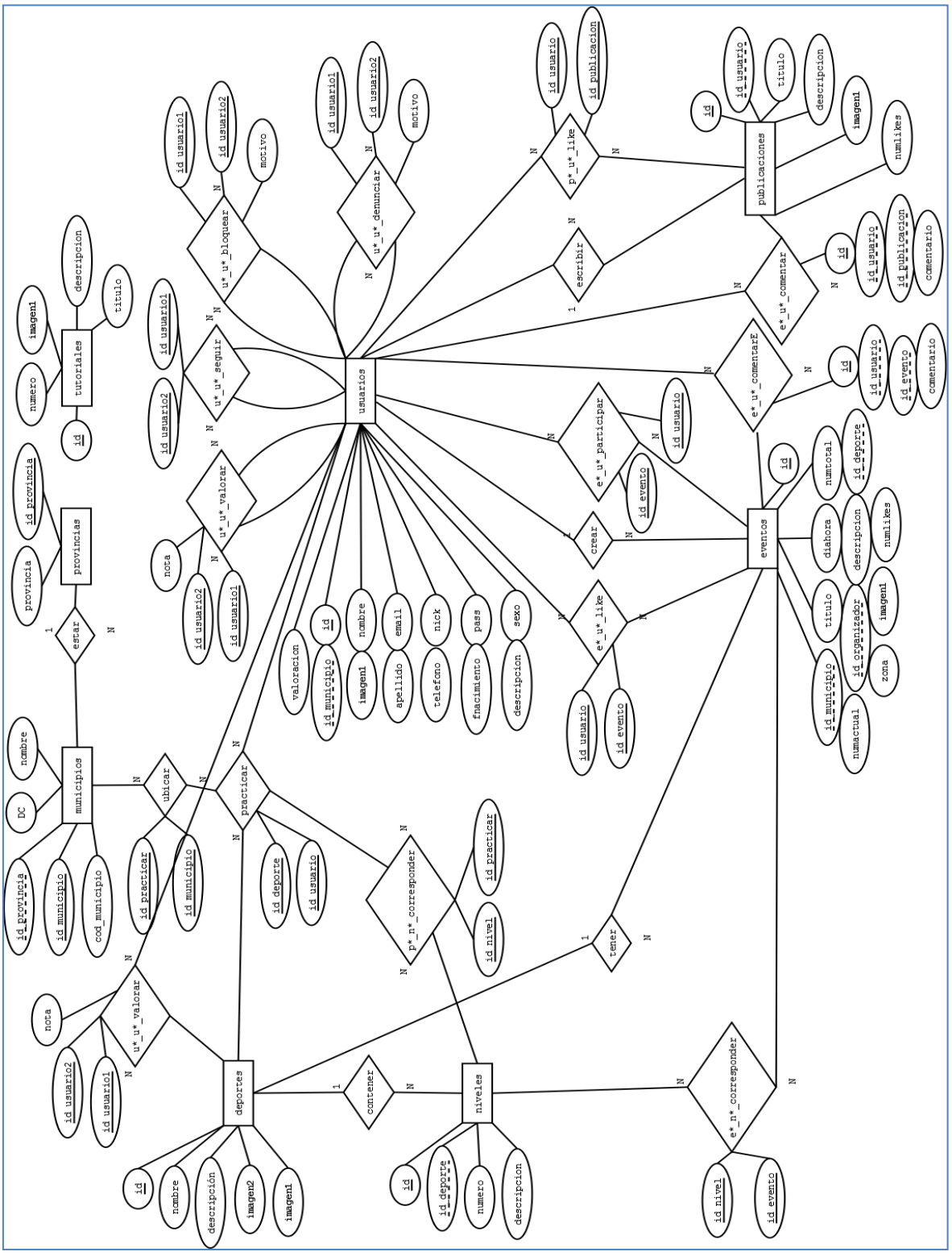

Figura 5.2: Diagrama E/R de la base de datos

Relaciones N-N

- **USUARIOS_DEPORTES_PRACTICAR:** es la relación encargada de almacenar los deportes en los que se inscribe cada usuario del sistema. Dicha relación es la más destacada de la base de datos ya que de la misma extienden otras dos relaciones.

- **PRACTICAR_NIVELES_CORRESPONDER:** representa una relación que vincula la relación anterior con la entidad que almacena los niveles. De esta forma, cada usuario que se inscribe en la práctica de un deporte, puede indicar los niveles que le corresponden de dicho deporte.

- **PRACTICAR_MUNICIPIOS_UBICAR:** vincula la relación *practicar* con la entidad que contiene todos los municipios del estado. Por consiguiente, cada usuario inscrito en un deporte, puede indicar los municipios en los que suele realizar la práctica deportiva pudiendo indicar diferentes municipios para cada deporte.

- **EVENTOS_USUARIOS_PARTICIPAR:** representa la relación de los usuarios que se han unido a un evento. El organizador del evento también se encuentra en esta tabla.

- **EVENTOS_NIVELES_CORRESPONDER:** es una relación de los niveles de práctica que se han elegido para un evento.

- **USUARIOS_USUARIOS_SEGUIR:** representa la relación entre usuarios que se siguen en la red. Un usuario puede seguir a otro sin que éste le siga.

- **DEPORTES_USUARIOS_VALORAR:** es una asociación de un usuario que realiza una valoración sobre un deporte en el que se encuentra inscrito.

- **EVENTOS_USUARIOS_VALORAR:** relación de un usuario con un evento que indica si a dicho usuario le gusta el evento.

- **PUBLICACIONES_USUARIOS_VALORAR:** es una relación de un usuario con una publicación que indica si el usuario está interesado en la misma.

- **USUARIOS_USUARIOS_BLOQUEAR:** representa una asociación de un usuario que bloquea a otro indicando un determinado motivo.

- **EVENTOS_USUARIOS_COMENTAR:** es una relación de un usuario con el evento sobre el que ha realizado un comentario.

- **PUBLICACIONES_USUARIOS_COMENTAR:** es una relación de un usuario con la publicación sobre la que ha realizado un comentario.

5.2.4 API RPC

Un principio muy importante de la aplicación es su comportamiento fuertemente interactivo. Para ello, se usa una técnica de desarrollo web llamada AJAX (*Asynchronous JavaScript And XML*) en la que se realizan llamadas al servidor mientras el usuario sigue navegando por la misma página web sin necesidad de recargar la misma. Las respuestas de estas llamadas por parte del servidor se realizan, entre otros formatos soportados, en JSON.

El servidor debe estar preparado para realizar estas funcionalidades. Por este motivo, se ha diseñado una API llamada RPC totalmente independiente de la parte de la aplicación, es decir, ambos módulos podrían estar en máquinas diferentes sin compartir recursos y el sistema seguiría funcionando correctamente. RPC (*Remote Procedure Call*) es un protocolo que permite ejecutar código en una máquina remota sin tener que preocuparse por las comunicaciones entre ambos. La gran ventaja de este diseño es que dicho módulo puede ser utilizado, no solo por la aplicación web desarrollada, sino por otro tipo de dispositivos con diferente soporte. El diseño interno del módulo sigue el patrón MC (modelo controlador) incorporando una capa de negocio intermedia entre ambos.

6 Implementación

6.1 Introducción

A continuación, se explica cómo se han implementado los diferentes componentes diseñados en el capítulo anterior. Se detalla el desarrollo de los módulos de mayor importancia indicando las herramientas utilizadas para su implementación.

6.2 Estructura del sistema

El sistema, como toda aplicación web, dispone de una parte cliente llamada front-end y de una parte servidora denominada back-end. Además, en cuanto a modularización, se dispone de una API RPC que dispone de la funcionalidad destacada del sistema y otro módulo que contiene la parte de la aplicación cómo vimos en la figura 5.1.

El front-end o interfaz es la parte que interactúa con los usuarios y su código corresponde a las vistas de la parte de la aplicación. Dichas vistas se han implementando en los lenguajes HTML, JS y CSS con la ayuda de Bootstrap y Backbone, frameworks de diseño web y de desarrollo JavaScript respectivamente. Por otra parte, el back-end o motor es la parte que procesa la entrada desde la interfaz y está formada por los controladores y modelos de la API y por los controladores del módulo de la aplicación. Se ha decidido implementar dicho motor en el lenguaje PHP ya que permite representar entidades como clases, está pensado para construir páginas web dinámicas y, además, está ampliamente extendido en desarrollo web.

Por último, destacar que la implementación del conjunto del sistema se ha realizado mediante el framework de desarrollo web PHP Zend. Esta herramienta cuenta con un gran número de utilidades como soporte para la conexión y manejo de la base de datos, diseño patrones de arquitectura MVC, interpretación de datos en formato JSON, solución de problema de seguridad como inyección SQL, etc. A continuación, se ha realizado un diagrama de clases UML que describe la estructura utilizada en la implementación del sistema.

6.2.1 Controladores de la parte de la aplicación

En la figura 6.1 se muestra la clase que contiene los controladores de la interfaz de la aplicación de usuarios y administradores. Dichos controladores, solicitados mediante peticiones HTTP, se pueden denominar 'vagos' ya que, únicamente, proporcionan las vistas a la parte del cliente y realizan la lectura y escritura de los parámetros de entrada/salida. Por requerimiento de Zend, el nombre del controlador corresponde a la página web que se quiere obtener.

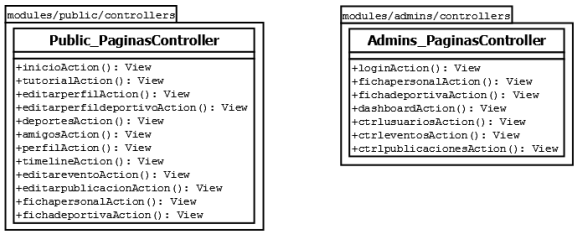

Figura 6.1: Diagrama de clases de los controladores de la parte de la aplicación

6.2.2 Conexión con la base de datos

El modelo es el encargado de realizar la conexión con la DDBB de la aplicación. Como se muestra en la figuras 6.2, 6.3, 6.4, el modelo está dividido en tres directorios:

- **Objects**: Este conjunto de clases son los contenedores de los objetos. Representan las entidades de la base de datos en la aplicación. Se podría prescindir de ellos pero sirven para saber de qué tipo de elemento son los resultados de los *select* y qué propiedades tienen los mismos. La clase padre *Mapper* contiene un conjunto de funciones comunes a todos los objetos.

- **DbTables**: Es un grupo de clases que hay que crear obligatoriamente para seguir la estructura propuesta por Zend para implementar el modelo. Contiene un único atributo con el nombre de la base de datos y heredan de la clase primitiva de Zend llamada *Zend_Db_Table*.

- **Mappers**: Son las clases que contienen las funcionalidades del modelo. Necesitan las clases *DbTable* para poder ejecutar operaciones en la DDBB como *insert, delete, select*, etc. y las clases *Objects* para poder devolver los resultados con el tipo correspondiente. Dichas clases mapean las acciones de la DDBB en la aplicación. Se ha creado una clase padre llamada *Mapper* que proporciona un conjunto de operaciones que, entre otras funcionalidades, permite realizar *selects* muy robustos

de una manera óptima. De esta forma, se consigue realizar diferentes *selects* con múltiples condiciones a través de una única función denominada *listAll*.

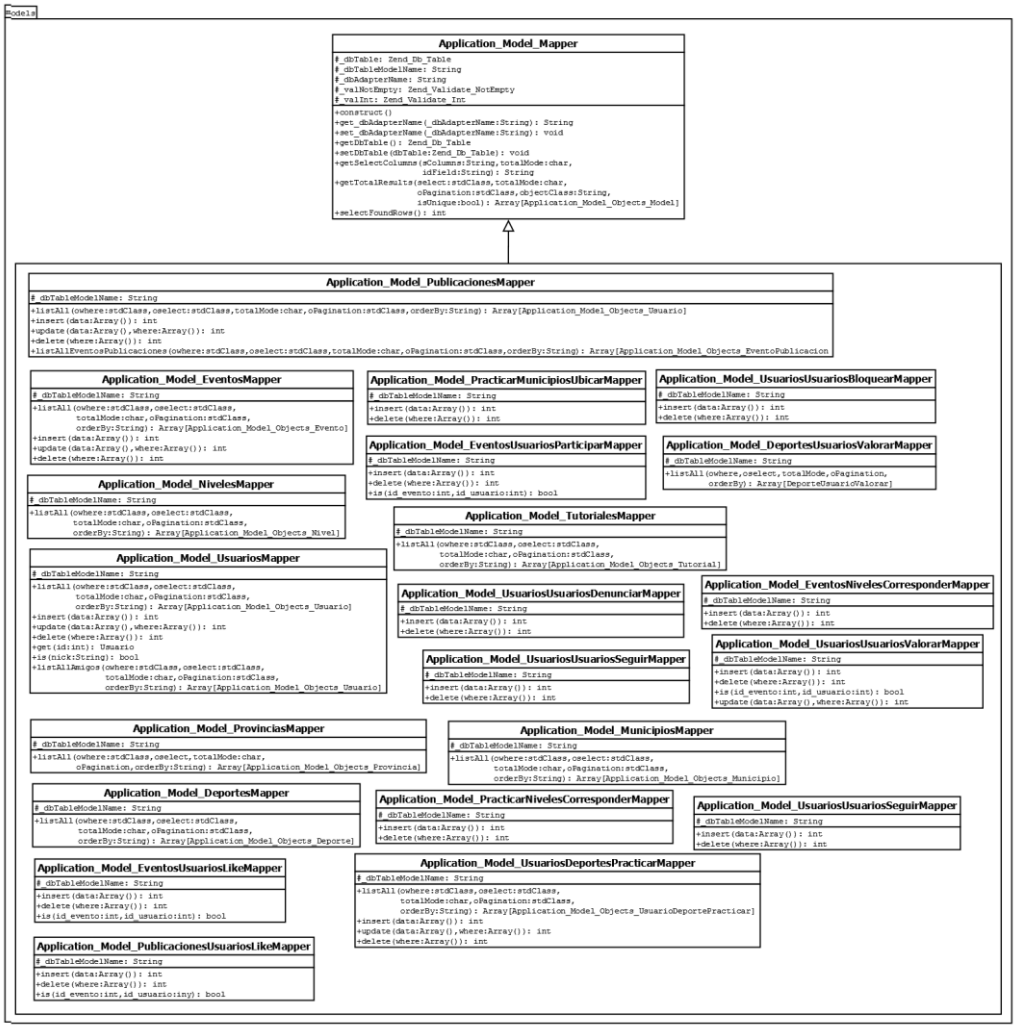

Figura 6.2: Diagrama de clases de los *mappers*

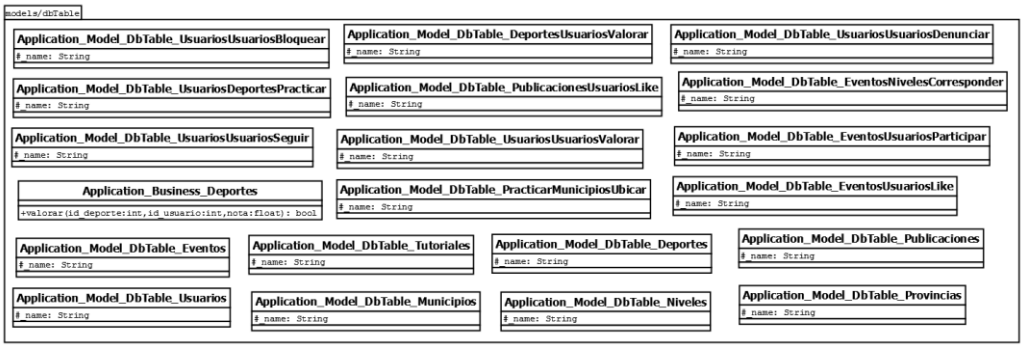

Figura 6.3: Diagrama de clases de los *dbTables*

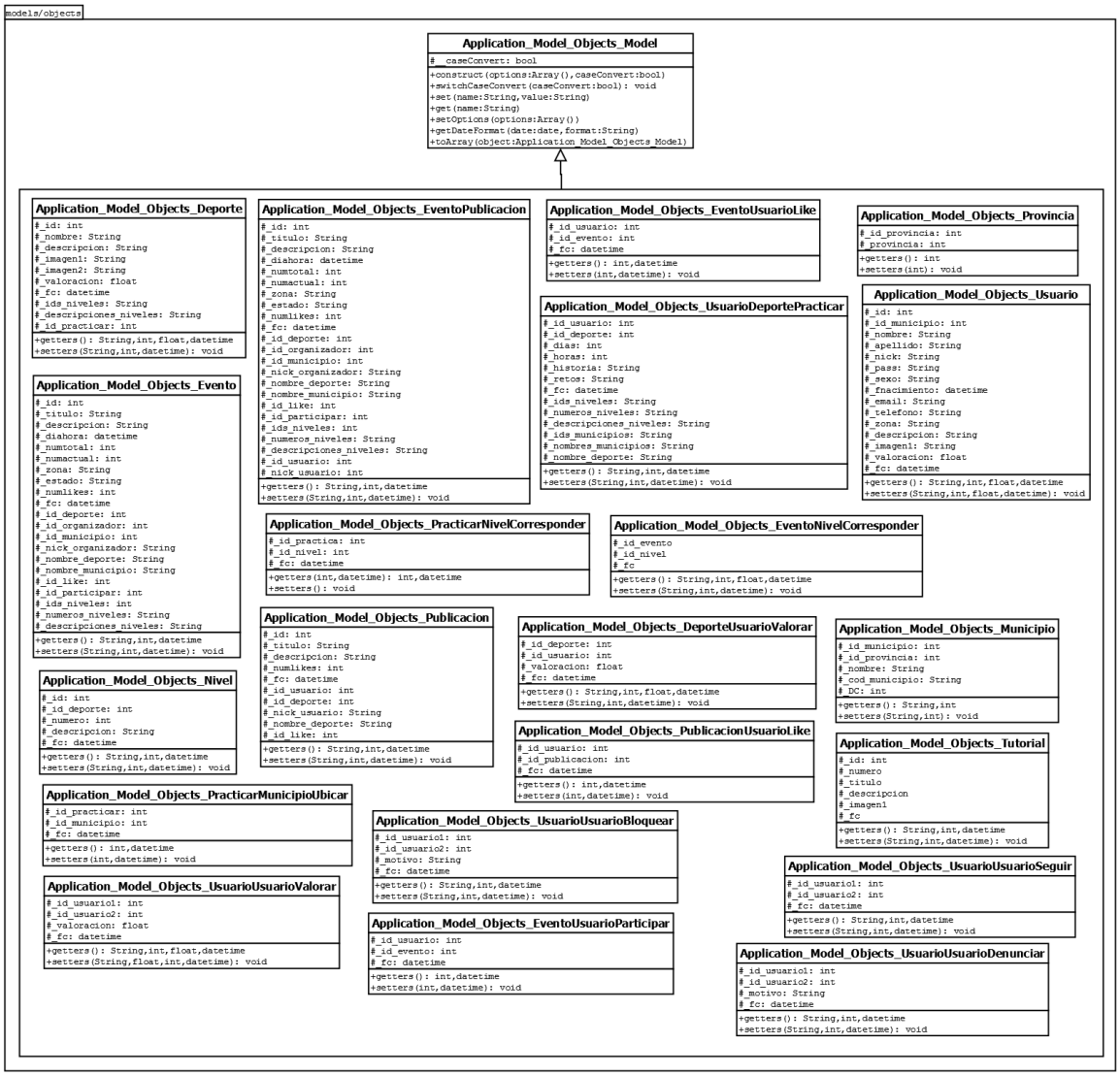

Figura 6.4: Diagrama de clases de los *objects*

6.2.3 Controladores de la API RPC

En la figura 6.5 se puede observar el diagrama de clases correspondiente a los controladores de la API RPC. Los controladores proporcionan la respuesta en formato JSON y, aunque no son restrictivos a otros protocolos como HTTP, nuestra aplicación cliente solicita los controladores mediante llamadas AJAX. Cada uno de sus métodos corresponde a acciones a realizar en el modelo o en la capa de negocio dotando de funcionalidad a la aplicación. El conjunto de clases heredan de una clase padre llamada *Services_RpcController* que es la encargada de realizar el envío de datos al cliente a través de la función *sendResponse*.

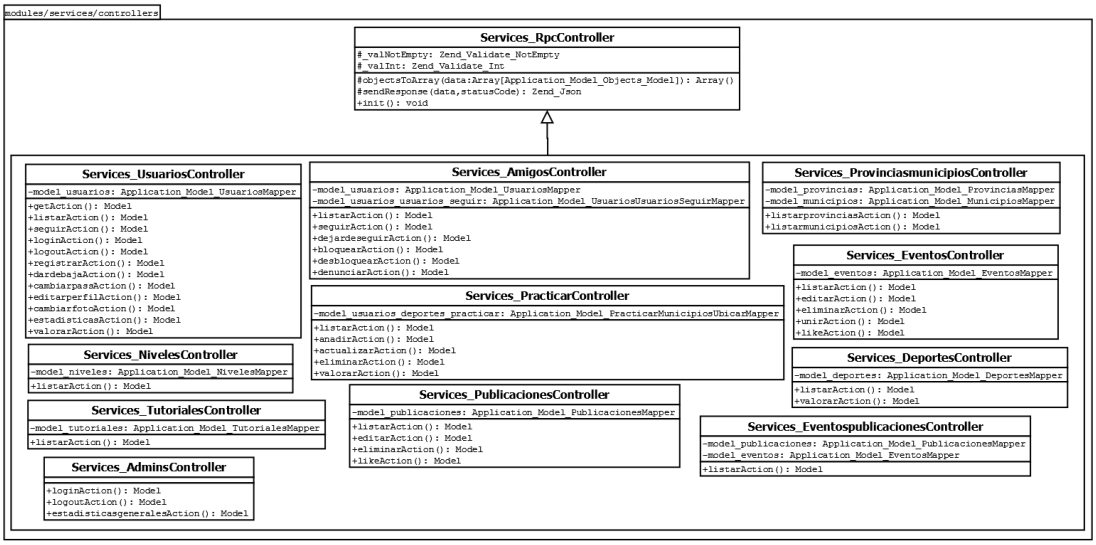

Figura 6.5: Diagrama de clases de los controladores de la API RPC

6.2.4 Capa de negocio

En la figura 6.6 se muestran las clases correspondientes a la capa de negocio de la aplicación utilizada para separar la lógica de negocio de la de diseño. Se utilizan dichas clases para programación de algoritmos y para operaciones que requieren más de un acceso a la base de datos. El nombre de la clase se corresponde con el controlador con el que está relacionado. La capa de negocio es una capa intermedia entre los controladores y el modelo de la API pero no es necesario que siempre se pase por ella para el intercambio de datos.

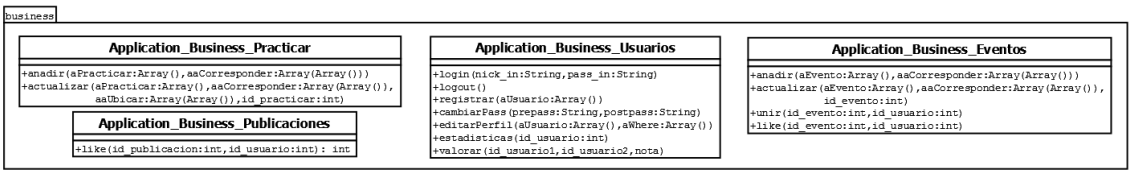

Figura 6.6: Diagrama de clases de la capa de negocio

6.3 Base de datos

La base de datos, mostrada en un diagrama entidad-relación en la figura 5.2 del capítulo anterior, se ha implementado mediante el sistema de gestión de bases de datos relacional MySQL. Para poder implementar dicha base de datos se ha realizado un proceso de normalización sobre la misma para transformar el modelo entidad-relación previamente diseñado en un modelo relacional.

6.4 Interfaz de usuario

La interfaz de esta aplicación está implementada para navegadores web y las vistas por las que está formada son ficheros HTML, JS y CSS: HTML permite mostrar el contenido de la página; JS permite implementar la interfaz de forma dinámica y, además, es la encargada de hacer la carga de datos obtenidos de la base de datos y de imágenes; CSS permite crear la presentación de la página. La interfaz se ha implementado con la ayuda del framework de diseño *Bootstrap 3* y del framework Javascript *Backbone*. Ambas herramientas se explican en detalle en el capítulo 4.

A continuación, se muestra y se realiza una breve descripción de las interfaces que componen el sistema en el rol de cliente:

6.4.1 Inicio de la aplicación

La página de inicio, mostrada en la figura 6.7, es la interfaz que se encuentra el usuario al acceder a la aplicación y contiene un breve resumen de los servicios del sistema. Los botones de la primera imagen permiten al usuario realizar el login o el registro en la app mediante un modal que se abre en la propia ventana. El tercer botón permite navegar a otra página que contiene una guía sobre la aplicación. Todas las páginas que conforman la web contienen una cabecera con diferentes opciones y un pie de página informativo.

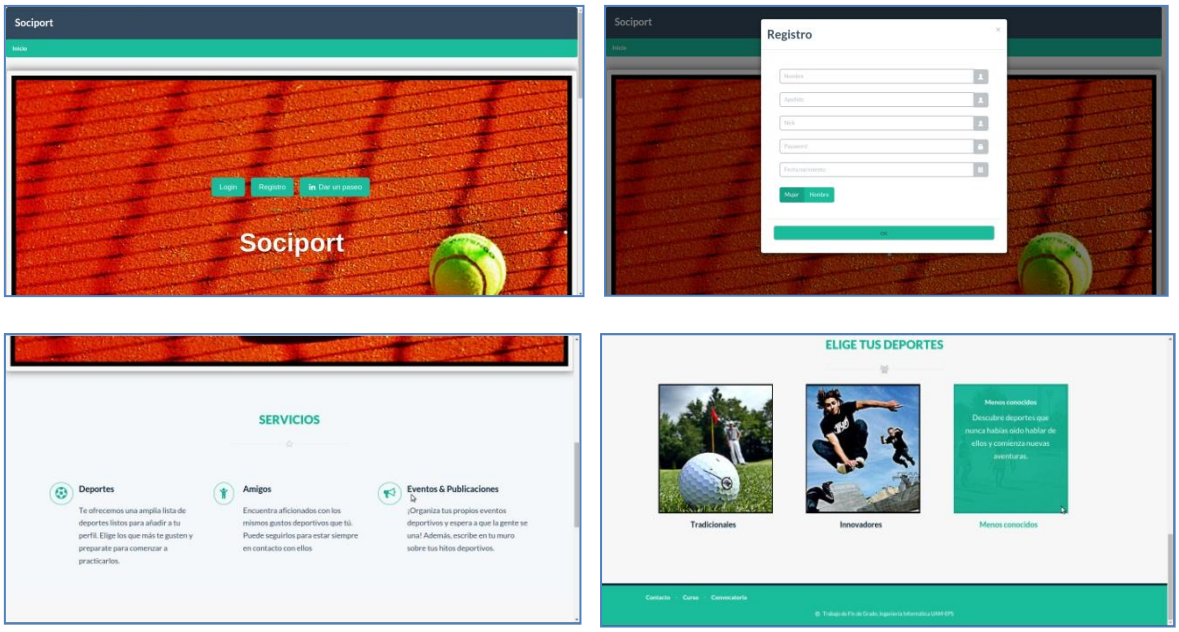

Figura 6.7: Interfaz de bienvenida

6.4.2 Interfaz de deportes

En la figura 6.8 se muestra la interfaz de deportes que contiene una lista con todos los deportes disponibles en la web. Se puede seleccionar cada deporte de la lista izquierda y se cargarán los datos del mismo en la vista de la mitad derecha de la página. Dichos datos están formados por una imagen, un cuadro con la descripción del deporte y una lista de checkboxes para elegir los niveles de práctica. En la parte inferior se encuentra un botón para añadir cada deporte al perfil del usuario.

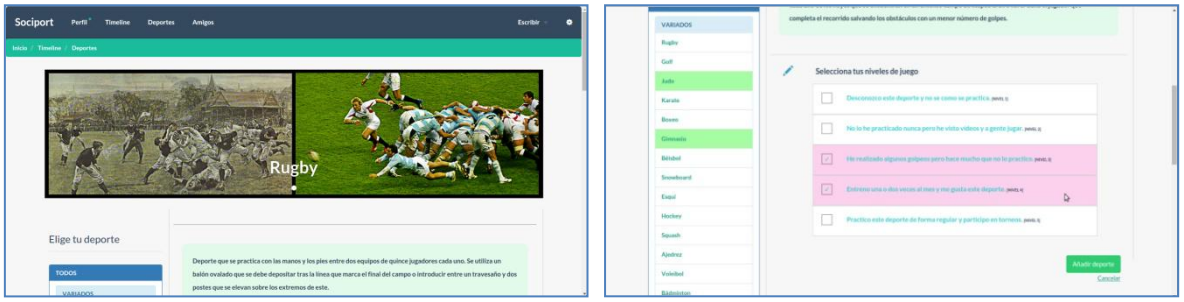

Figura 6.8: Interfaz de deportes

6.4.3 Interfaz de edición de perfiles

Edición de perfil personal

Al registrarse en la aplicación el usuario tiene que proporcionar una serie de datos personales obligatorios. Una vez registrado y logueado, el cliente tiene la opción de modificar dichos datos y, además, puede proporcionar unos datos adicionales para completar su perfil personal. Para ello, se emplea un conjunto de inputs, selects, textboxes y un cuadro para cambiar la imagen de perfil. En la figura 6.9 se muestra dicha página:

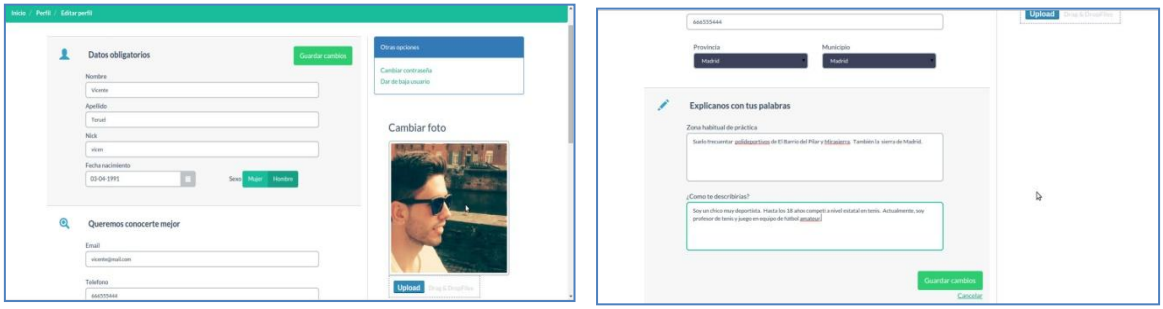

Figura 6.9: Interfaz de edición de perfil personal

Edición de perfil deportivo

Por cada deporte en el que se inscribe un usuario se pueden proporcionar una serie de datos adicionales como municipios, horas y días de practica, etc. además de modificar los niveles de practica elegidos al añadir el deporte. Para ello, como se aprecia en la figura 6.10, se dispone de una vista a la izquierda de la imagen con la lista de deportes añadidos y a la derecha un conjunto de checkboxes, selects y texboxes para rellenar dicha información.

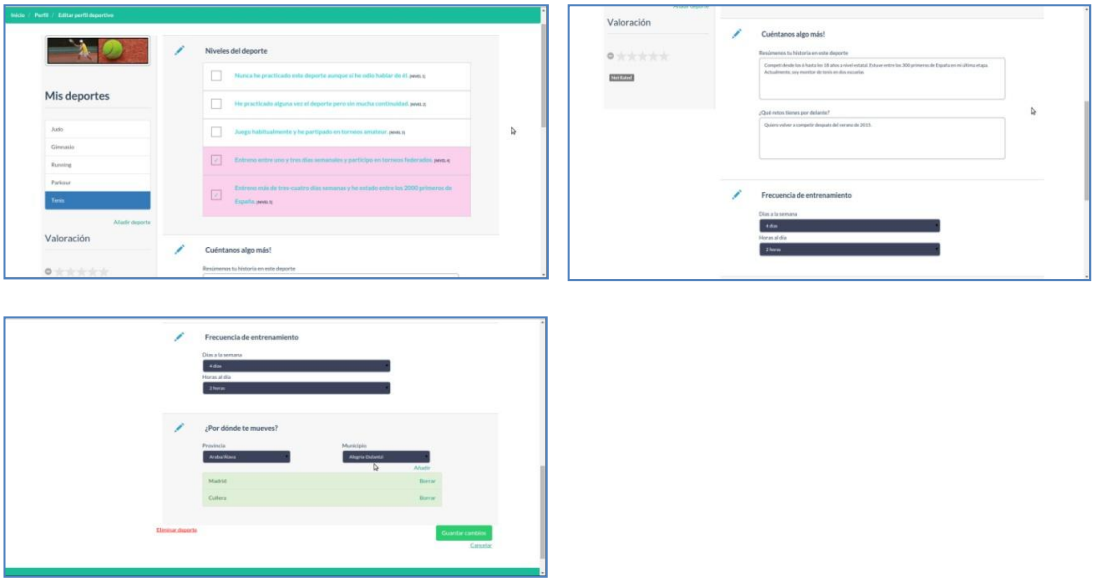

Figura 6.10: Interfaz de edición de perfil deportivo

6.4.4 Interfaz de eventos y publicaciones

Eventos

Para crear nuevos eventos deportivos o para modificar uno ya creado se dispone de la interfaz mostrada en la figura 6.11. Dicha interfaz está formada por elementos de entrada de datos similares a los explicados en las anteriores interfaces.

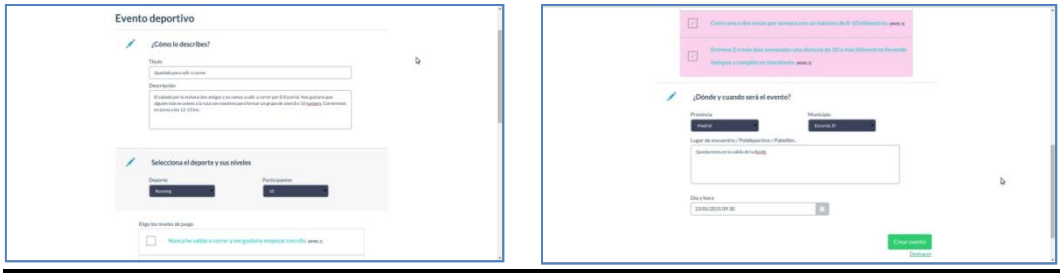

Figura 6.11: Interfaz de eventos

Publicaciones

Las publicaciones, al igual que los eventos, utilizan la misma interfaz para crear una nueva publicación que para modificar una ya generada. En la figura 6.12 se muestra la implementación de dicha interfaz:

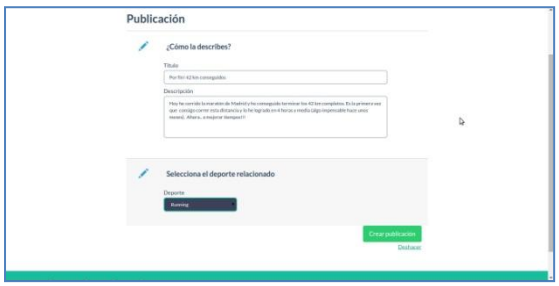

Figura 6.12: Interfaz de publicaciones

6.4.5 Interfaz de fichas personales y deportivas

Para poder ver de forma detallada la información personal y deportiva de otros usuarios se dispone de dos interfaces con contenido informativo. Están formadas únicamente por tablas y, en el caso de la ficha deportiva, la primera tabla contiene enlaces a tablas inferiores de cada deporte. En la figura 6.13 se muestran las dos interfaces:

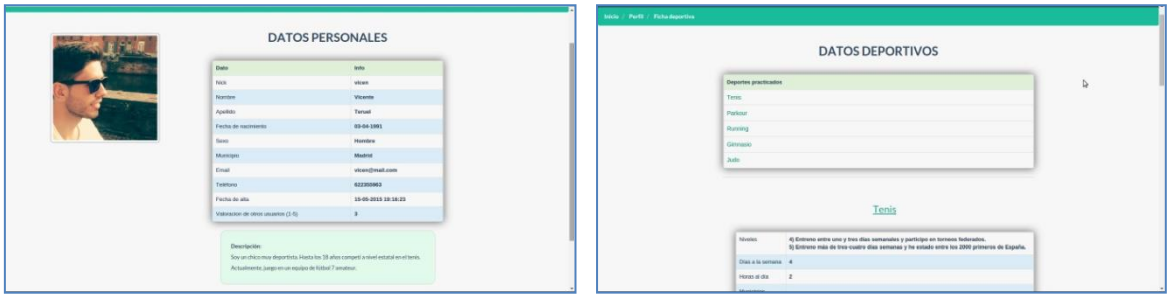

Figura 6.13: Interfaz de ficha personal y deportiva

6.4.6 Interfaz de amistades

La página de amistades está formada por un *nav* en el que se puede elegir el tipo de amigos a listar en la vista inferior. Cada amigo está formado por un cuadro con una imagen más botones de opciones. En la parte inferior se dispone de un paginador para ir mostrando los amigos en una cantidad definida previamente y así no generar una lista muy alta de datos. En la figura 6.14 se puede ver el resultado:

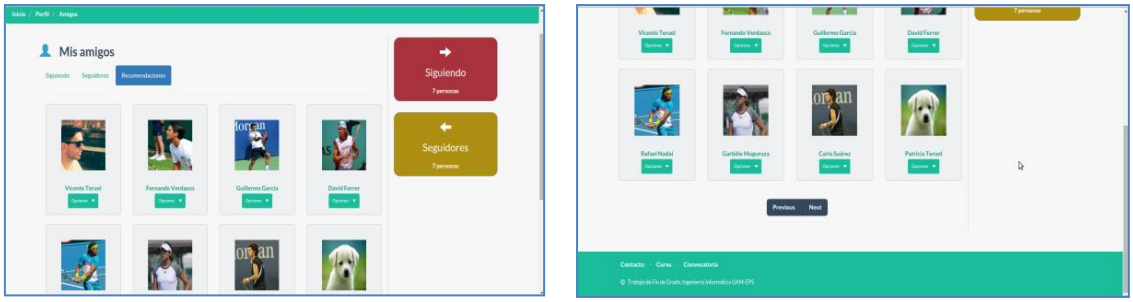

Figura 6.14: Interfaz de amistades

6.4.7 Interfaz del perfil de usuario

La interfaz perfil es la página principal de cada usuario. En ella se encuentran enlaces al resto de páginas de la web como editar perfiles, amigos, fichas, etc. Entre sus componentes destacamos la imagen del usuario, un *nav* acompañado de un *select* para filtrar el listado de los eventos y publicaciones del propio usuario mediante un paginador (la lista generada será muy extensa; en la figura 6.15 es un caso de prueba). Si el usuario que accede al perfil es el propio usuario se verán todas las opciones disponibles pero si es el perfil de otro usuario no estarán disponibles algunas opciones como editar perfiles, borrar contenidos, etc.

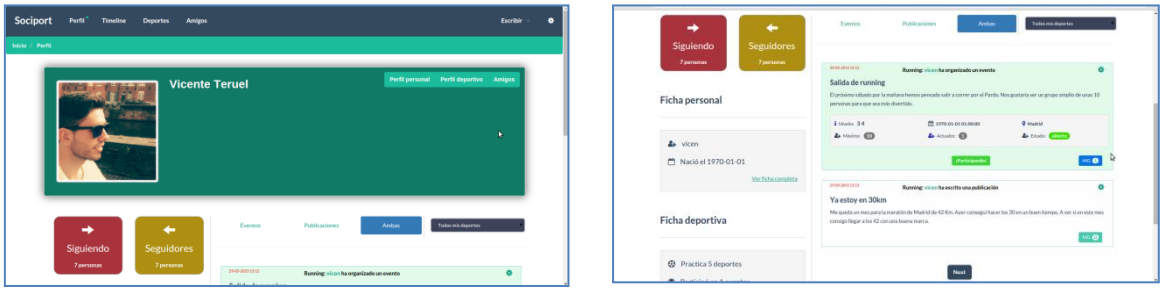

Figura 6.15: Interfaz de perfil de usuario

6.4.8 Interfaz del timeline

La interfaz timeline es muy similar al filtrado de eventos y publicaciones explicados en el punto anterior para el perfil del usuario. La diferencia es que en este listado se encuentran los eventos y publicaciones del resto de usuario de la comunidad: en primer lugar los usuarios que se encuentre siguiendo el propio usuario y después el resto de usuarios. Está formado por un *nav* y un *select* para el filtrado y por los cuadros de eventos y publicaciones correspondientes.

Se puede comprobar también como están formados los cuadros de eventos y publicaciones. Ambos están compuestos por un título, una descripción, botón de *MG* y, en el caso de los eventos, de un cuadro interior con información detallada con múltiples despliegues y un botón para unirse al evento. En la figura 6.16 se puede ver dicha interfaz:

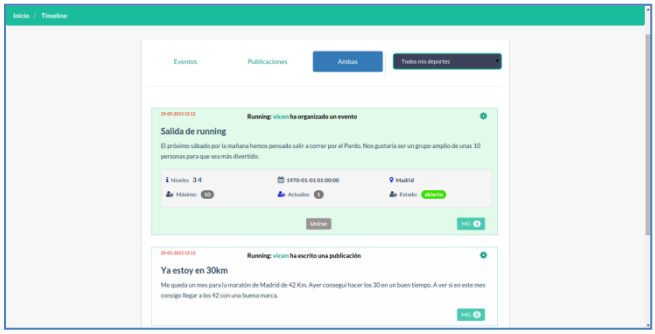

Figura 6.16: Interfaz del timeline

6.4.9 Otras interfaces

Hay una página que contiene una guía de aprendizaje de cómo utilizar la aplicación. Está formada por *carousel* que contiene capturas del sistema y un cuadro en el que se describe cada sección.

Además, como se ha visto en alguna de las anteriores interfaces, se dispone de un cuadro de estrellas para realizar valoraciones de deportes. En la figura 6.17 se puede ver la interfaz de valoraciones.

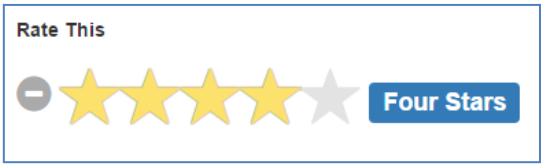

Figura 6.17: Interfaz de valoración

En los siguientes apartados se muestran las interfaces del rol administrador y se realiza una breve explicación de las mismas:

6.4.10 Login de administrador

La página de login, mostrada en la figura 6.18, es la interfaz que se encuentra el encargado de mantener la red al acceder a la aplicación. Está formada por un formulario para autenticar a un administrador del sistema.

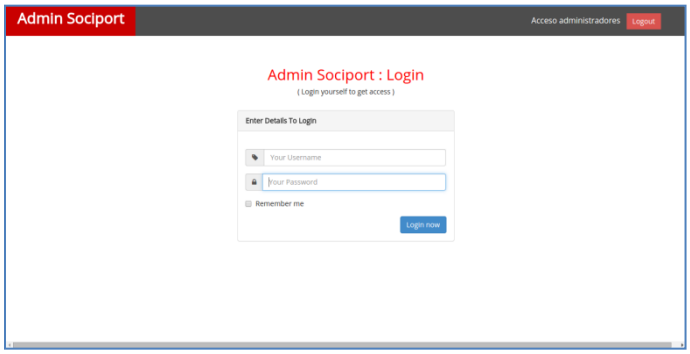

Figura 6.18: Interfaz de login de administrador

6.4.11 Dashboard

La interfaz dashboard es la página principal de la aplicación para administradores. Contiene un resumen de uso de la aplicación y una lista de opciones para acceder a los siguientes servicios: control de usuarios, control de eventos y control de publicaciones. En la figura 6.19 se puede ver la interfaz:

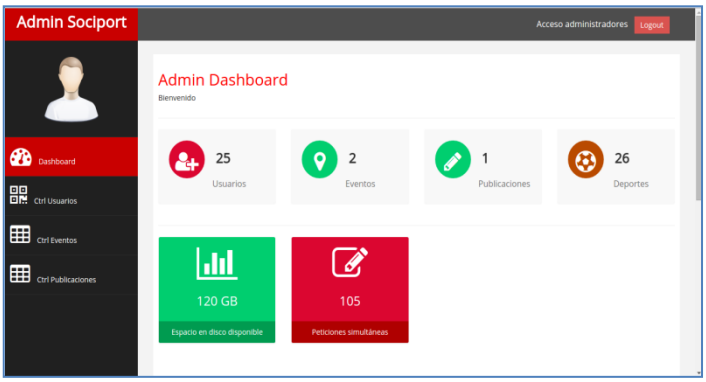

Figura 6.19: Interfaz de dashboard

6.4.12 Interfaces de control

Control de usuarios

En la interfaz de control de usuarios, mostrada en la figura 6.20, se listan los clientes registrados en el sistema. Se utiliza una tabla ordenada por fecha de actualización que contiene la id del usuario y dos links que acceden a las fichas personales y deportivas del usuario. Dichas fichas son similares a las mostradas en la figura 6.13. Se tiene la opción de eliminar el usuario desde la propia tabla.

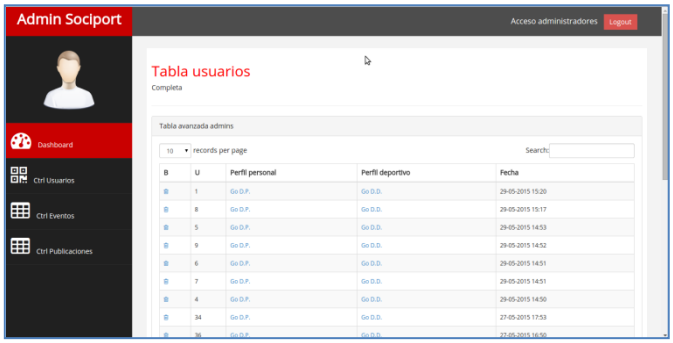

Figura 6.20: Interfaz de control de usuarios

Control de eventos y publicaciones

Las interfaces de control de eventos y publicaciones utilizan una tabla ordenada por fecha de actualización para mostrar el título y descripciones que el usuario introduce en cada uno de ellos. El administrador tiene la opción de eliminar las entradas de la tabla individualmente. En la siguiente figura se muestran las interfaces:

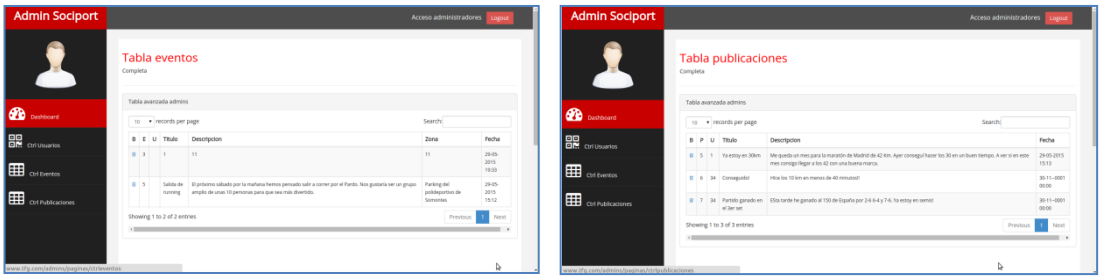

Figura 6.21: Interfaces de control de eventos y publicaciones

7 Validación y Verificación

7.1 Verificación

El objetivo de realizar una verificación del sistema es comprobar que todos los componentes que forman el sistema funcionan correctamente. Para llevar a cabo este proceso se ha creado un plan de pruebas formado por pruebas unitarias, de integración, de interfaz y de inspección de código.

7.1.1 Estrategia de pruebas

En el sistema hay tres componentes claramente diferenciados: la base de datos, la parte de aplicación y la API RPC. Por este motivo, se han creado planes específicos para cada una de estas partes. A continuación, se detallan las estrategias seguidas:

Revisión de código

El método de reconocimiento de código consiste en una lectura del mismo una vez finalizada la programación para detectar posibles errores a simple vista. Dicho método se ha utilizado para revisar cada una de las clases que forman los controladores de la parte de la aplicación y los controladores y modelos de la API RPC. Los métodos que han requerido de una revisión más minuciosa han sido los *listAll* de las clases del modelo ya que contienen un número muy alto de filtros para hacerlos reutilizables y la única *query* que se genera alcanza un tamaño considerable.

Pruebas unitarias

Las pruebas unitarias son las encargadas de comprobar cada módulo individualmente. Este tipo de prueba se ha realizado sobre la DDBB y el módulo RPC para determinar que los modelos y controladores funcionan correctamente y no tienen comportamientos erróneos. Los planes de pruebas unitarias de caja blanca y caja negra se han realizado con la herramienta DevTools disponible para Google Chrome.

Pruebas de integración

Tras realizar las pruebas unitarias se ha iniciado el plan de pruebas de integración. Estas pruebas tienen el objetivo de probar el conjunto de módulos que forman el sistema una vez unidos en bloque. Se ha realizado un plan escalable: en primer lugar, se ha comprobado el correcto funcionamiento de la parte cliente con el módulo de la aplicación, de la parte cliente con la API RPC, de la base de datos con los modelos de la API y de la parte de aplicación con la propia API; finalmente, se han realizado tests integrando todos los módulos.

Pruebas de interfaz gráfica

El objetivo de las pruebas de interfaz gráfica es comprobar que la visualización de cada una de las páginas web que forman el sistema se realiza de forma correcta y que todos los botones, elementos de entrada de datos, etc. funcionan de la forma esperada. Al tratarse de una página web con un alto contenido dinámico también se ha comprobado si la visualización y cambio de contenidos a partir de acciones de usuarios se realiza de forma óptima.

7.1.2 Desarrollo de las pruebas

En este apartado se describe cómo y cuándo se han realizado las pruebas detallando el procedimiento seguido y los resultados obtenidos para cada una de las estrategias anteriores.

Revisión de código

La inspección de código se ha realizado después de programar cada una de las clases PHP del servidor. Al ser un lenguaje interpretado y no disponer de una fase de compilación, esta prueba adquiere mayor relevancia. El estudiante ha realizado una lectura minuciosa del código en la que se han detectado los siguientes fallos:

- Errores en la sintaxis de llamadas entre los métodos de los controladores *listar* y de los modelos *listAll*. Las funciones que hacen *selects* a la DDBB trabajan con los siguientes parámetros: oselect, owhere, opagination, totalMode y orderBy. Se

detectaron fallos en la escritura y posterior lectura de los parámetros oselect y owhere de tipo *stdClass* en cuatro de estos métodos.

- Errores sintácticos en las *queries* realizadas en los métodos *listAll* del modelo. Se han detectado errores en la programación por las condiciones que impone Zend para realizar el modelaje de los *selects* en los mappers de eventos, publicaciones y usuarios_practicar_niveles.

Pruebas unitarias

Las pruebas de caja blanca se han realizado sobre los métodos listar de los controladores de la API RPC y de sus correspondientes mappers. El único error detectado en la prueba de los caminos independientes ha sido el siguiente:

- Se realizaban más de un join a la base de datos con el mismo alias para las tablas usuarios, eventos_usuarios_like y publicaciones_usuarios_like.

Las pruebas de caja negra se han elaborado sobre los controladores de la API RPC y de la base de datos. Se han introducido valores de entrada coherentes y otros límites para comprobar la salida de los mismos. A continuación se describen los resultados:

- Las variables de tipo *datetime* no se almacenaban correctamente en la base de datos debido al formato establecido. Se ha optado por realizar el cambio de formato en los mappers.

- Se desbordaban *strings* en la base de datos ya que no se controlaba en el servidor el tamaño máximo disponible. Se ha comprobado el tamaño máximo en los controladores de la API.

Pruebas de integración

Una vez acabado la programación de los módulos y haber pasado cada uno de ellos las pruebas unitarias, se han realizado las pruebas de integración. La integración de la parte del cliente con la parte de aplicación y de la API RPC con la DDBB no ha generado errores destacables.

Por otra parte, la integración de la parte del cliente con la API RPC sí que ha detectado fallos relevantes en el paso de información entre cliente-servidor ya que la estructura utilizada en los controladores amigos, usuarios, eventos y publicaciones no era la misma que la estructura seguida en los modelos y colecciones de Backbone de la parte del cliente.

Finalmente, se ha detectado fallos en la comunicación realizada con Zend_Http_Client entre los dos módulos servidores para la realización del login en la parte de la aplicación ya que la llamada no estaba bien definida.

Pruebas de interfaz gráfica

Las últimas pruebas que se han realizado han sido las de interfaz gráfica. En primer lugar, el estudiante ha llevado a cabo un plan para intentar probar todas las posibles acciones de la interfaz. Se detectaron fallos en las cargas dinámicas de las páginas *editarperfildeportivo* y *editarperfilpersonal* y en las dimensiones de la imagen de *perfil* de usuarios.

7.2 Validación

El objetivo de validar el sistema es comprobar que los requisitos funcionales y no funcionales del capítulo 5 se cumplen satisfactoriamente.

7.2.1 Estrategia y desarrollo de validación

Para validar el sistema se han comprobado si los requisitos previamente definidos se cumplían satisfactoriamente desde el punto de vista de un usuario final. La estrategia seguida por el estudiante ha sido la de colaborar con usuarios ajenos al desarrollo de la aplicación para que naveguen por la web como usuarios finales. Dichos usuarios disponían de conocimientos diferentes sobre navegación web, desde estudiantes de la Escuela hasta personas adultas que no suelen navegar por la red.

En primer lugar, el estudiante introdujo usuarios en la red ficticios y creó una serie de eventos y publicaciones para que los usuarios que iban a probar la aplicación tuvieran otros usuarios anteriores con los que encontrarse. A cada usuario se le dio un tiempo de 20-25 minutos para interactuar con la aplicación, independientemente de sus conocimientos.

Durante el conjunto de pruebas el estudiante disponía de dos tablas: en la primera se iban tachando los requisitos que se cumplían correctamente y, en la segunda, se anotaban los fallos encontrados por los usuarios y el requisito con el que estaba relacionado dicho error.

Los resultados obtenidos en esta primera vuelta han sido los siguientes:

- Requisitos funcionales: 19 de 21 cumplidos.
- Requisitos no funcionales: 6 de 7 cumplidos.

Tras realizar la corrección de los tres requisitos que no cumplían se volvió a realizar la misma prueba con los mismos usuarios pero reduciendo el tiempo de la prueba a 10-15 minutos ya que los usuarios ya habían adquirido conocimientos sobre el funcionamiento de la web.

Los resultados finales de la segunda vuelta fueron satisfactorios y todos los requisitos pasaron a la primera tabla:

- Requisitos funcionales: 21 de 21 cumplidos.
- Requisitos no funcionales: 7 de 7 cumplidos.

8 Evaluación

8.1 Evaluación de los usuarios

Usuarios con diferente nivel de navegación web han evaluado el sistema tras su validación. Todos ellos coinciden que el objetivo por el que se propuso este trabajo de fin de grado se ha cumplido de forma satisfactoria.

Se ha considerado muy útil la usabilidad que aporta el sistema en comparación con otras herramientas con objetivos similares. Han señalado la facilidad con la que se pueden crear eventos y publicaciones y modificar los mismos sin necesidad de tener experiencia de uso con la aplicación. El cuadro de evento y el despliegue de su información ha sido el punto más destacado en cuanto a utilidad.

La interfaz gráfica ha sido evaluada favorablemente ya que todos han recalcado la facilidad de navegación que ofrece la web. En pocos minutos los usuarios fueron capaces de utilizar todos los servicios ofrecidos de forma intuitiva.

El sistema de recomendación por estrellas de los diferentes deportes ha sido señalado por los usuarios como una forma fácil y directa de conocer si el deporte ofertado puede ser de utilidad para ellos. Además, las descripciones de los deportes, las definiciones de sus niveles y las imágenes aportadas han sido destacadas por los usuarios probadores.

Finalmente, han aportado la posibilidad de crear un módulo de mensajería privada o de comentarios para facilitar la comunicación directa entre los usuarios del sistema. Dicha ampliación ha sido nombrada en las líneas futuras de trabajo.

8.2 Beneficios

El proyecto realizado en este TFG permite a los usuarios relacionarse entre sí con gustos deportivos en común a través de una sencilla herramienta web. Utilizar una aplicación para dar a conocer deportes tradicionales, innovadores, etc. a la mayor cantidad

de usuarios posibles puede ser un gran incentivo para esa parte de la población que quiere realizar actividades deportivas.

El diseño estructurado del sistema permite separar de forma óptima diferentes módulos de la web para poder ser reutilizados con otros dispositivos con diferente soporte. La API RPC permite disponer de las funcionalidades del sistema en un conjunto de bibliotecas independientes de la parte de aplicación.

Uno de los beneficios más destacados del sistema es la importante carga dinámica de datos que se realiza. Se abandona la tradicional forma de implementar una página web con código PHP embebido en código HTML que genera una página web con alto contenido estático. De esta forma, con la ayuda de un framework Javascript como Backbone, se inscrusta código JS en los HTML para permitir realizar cargas dinámicas del contenido y dotar de mayor flexibilidad a la web.

9 Conclusiones y líneas futuras

9.1 Conclusiones

En este Trabajo de Fin de Grado se ha realizado un proceso de Ingeniería del Software para el desarrollo de una aplicación web. El trabajo, realizado íntegramente por el estudiante, ha consistido en el diseño e implementación de una herramienta cuyo objetivo es unir a aficionados al deporte con gustos en común para facilitar la práctica del mismo.

En el estudio del estado del arte se ha detectado que los sistemas que se encuentran operativos en el mercado no están preparados para ser utilizados por usuarios con diferentes niveles de conocimientos sobre navegación web. Por este motivo, se ha diseñado un sistema con una interfaz gráfica sencilla, interactiva y usable para un amplio rango de usuarios. Una vez terminada la construcción del sistema, se aprecia que los requisitos establecidos al comienzo del proyecto se han cumplido satisfactoriamente.

Este trabajo ha proporcionado al estudiante multitud de conocimientos de programación web debido al uso de varias herramientas de desarrollo, ampliamente extendidas en empresas de desarrollo web, que eran completamente nuevas para el mismo y que ha necesitado realizar un aprendizaje desde cero. Por ello, el tiempo estipulado para la realización del proyecto ha superado las 300 horas de trabajo.

9.2 Líneas futuras

Tras el desarrollo del proyecto se ha obtenido una primera versión del sistema totalmente operativa y que contiene la funcionalidad básica del mismo. Se han definido un conjunto de ampliaciones para mejorar la interacción del usuario con el sistema y de algunas de sus funcionalidades.

A medio plazo se han propuesto las siguientes tres mejoras del sistema:

- Incorporar un servicio de comentarios para eventos y publicaciones. Los usuarios podrían expresar sus opiniones sobre diversos contenidos, estrechando el lazo entre usuarios.

- Incluir niveles de privacidad a los usuarios para que puedan elegir el grupo de la comunidad que puede interactuar con ellos. Además, se incluirían opciones de bloqueo y denuncia para el usuario.

- Ampliar el número de deportes ofertados en el catálogo.

Por otra parte, se han definido una serie de ampliaciones de la aplicación para ser realizadas a largo plazo:

- Creación de un módulo de mensajería privada entre usuarios para facilitar la comunicación entre los usuarios de la comunidad.

- Incluir un buzón de sugerencias. Los usuarios podrán enviar nuevas actividades deportivas a los administradores de la web para que sea estudiada la posible incorporación de dichas actividades a la lista de deportes disponibles del sistema.

- Crear una sección de deportes para personas con diversidad funcional. La web pretende acercar el mundo del deporte al mayor número de usuarios posibles y que mejor forma para hacerlo que incluir deportes para un sector que no dispone de tantas herramientas en internet con este fin. Para llevar a cabo esta mejora habría que adaptar la web para que se cumpliesen las 'Pautas de Accesibilidad para el Contenido Web (WCAG) 2.0' cumpliendo, al menos, con el nivel de conformidad A, es decir, disponer de una versión alternativa mínima [27].

10 Referencias

[1] Camilo López, «Timpik», [En línea]. Available: www.timpik.com/. [Último acceso: 2015].

[2] Decathlon, «Decathlon Sport Meeting», [En línea]. Available: https://sportmeeting.decathlon.com/. [Último acceso: 2015].

[3] Xporty «Xporty», [En línea]. Available: https://xporty.com/. [Último acceso: 2015].

[4] PortalTIC, «Depormeet, una nueva red social para amantes del deporte» *Europapress*, 18 Abril 2013.

[5] SocialDrive, «SocialDrive», [En línea]. Available: www.socialdrive.es/. [Último acceso: 2015].

[6] Filmaffinity, «Filmaffinity», [En línea]. Available: www.filmaffinity.com/. [Último acceso: 2015].

[7] Mark Zuckerberg, «Facebook», [En línea]. Available: https://es-es.facebook.com/. [Último acceso: 2015].

[8] Hugo Fernando Arboleda Jiménez, «Modelos De Ciclo De Vida En Desarrollo De Software En El Contexto De La Industria Colombiana De Software», *Revista Ingenierías*, 2005.

[9] Oracle, «MySQL», [En línea]. Available: https://www.mysql.com/. [Último acceso: 2015]

[10] Oracle, «MySQL Workbench», [En línea]. Available: https://www.mysql.com/products/workbench/. [Último acceso: 2015]

[11] The PHP Group, «PHP Documentation», [En línea]. Available: php.net/. [Último acceso: 2015].

[12] W3C, «HTML5», [En línea]. Available: www.w3.org/TR/html5/. [Último acceso: 2015].

[13] W3C, «Style Sheets», [En línea]. Available: www.w3.org/TR/html401/present/styles.html. [Último acceso: 2015]

[14] Douglas Crockford, «The World's Most Misunderstood Programming Language», [En línea]. Available: http://javascript.crockford.com/javascript.html. [Último acceso: 2015]

[15] The jQuery Foundation, «Jquery», [En línea]. Available: http://jquery.com/. [Último acceso: 2015].

[16] Zend Technologies, «Zend», [En línea]. Available: framework.zend.com/. [Último acceso: 2015].

[17] Backbone, «Backbone.js», [En línea]. Available: backbonejs.org/. [Último acceso: 2015].

[18] Backbone, «Backbonetutorials», [En línea]. Available: https://backbonetutorials.com/. [Último acceso: 2015].

[19] Bootstrap, «Bootstrap is the most popular HTML, CSS, and JS framework for developing responsive, mobile first projects on the web», [En línea]. Available: getbootstrap.com/. [Último acceso: 2015].

[20] Sphinx, «bootstrap-datepicker », [En línea]. Available: bootstrap-datepicker.readthedocs.org/en/latest/. [Último acceso: 2015].

[21] Eonasdan, «Bootstrap 3 Datepicker v4 Docs», [En línea]. Available: eonasdan.github.io/bootstrap-datetimepicker/. [Último acceso: 2015].

[22] Designmodo, «Free User Interface Kit», [En línea]. Available: designmodo.github.io/Flat-UI/. [Último acceso: 2015].

[23] D. Gandy, «Font Awesome, the iconic font and CSS toolkit», [En línea]. Available: http://fortawesome.github.io/Font-Awesome/. [Último acceso: 2015].

[24] Binarycart, «Free Bootstrap admin template», [En línea]. Available: binarycart.com/. [Último acceso: 2015].

[**25**] Kartik Visweswaran, «Bootstrap Star Rating», [En línea]. Available: http://plugins.krajee.com/star-rating. [Último acceso: 2015].

[**26**] The Apache Software Foundation, «Apache 2» . [En línea]. Available: http://httpd.apache.org/. [Último acceso: 2015].

[27] W3, «Web Content Accessibility Guidelines (WCAG) », [En línea]. Available: www.w3.org/WAI/intro/wcag/. [Último acceso: 2015].

Bibliografía complementaria

Tom Hughes-Croucher & Mike Wilson, «*Node: Up and Running*», O'Reilly Media, 2012.

Garann Means, «*Node for Front-End Developers*», O'Reilly Media, 2012.

Sandeep Kumar Patel, «*Larger Cover Developing Responsive Web Applications with AJAX and jQuery*», Packt Publishing, 2014.

Josh Lockhart, «*Modern PHP*», O'Reilly Media, 2015.